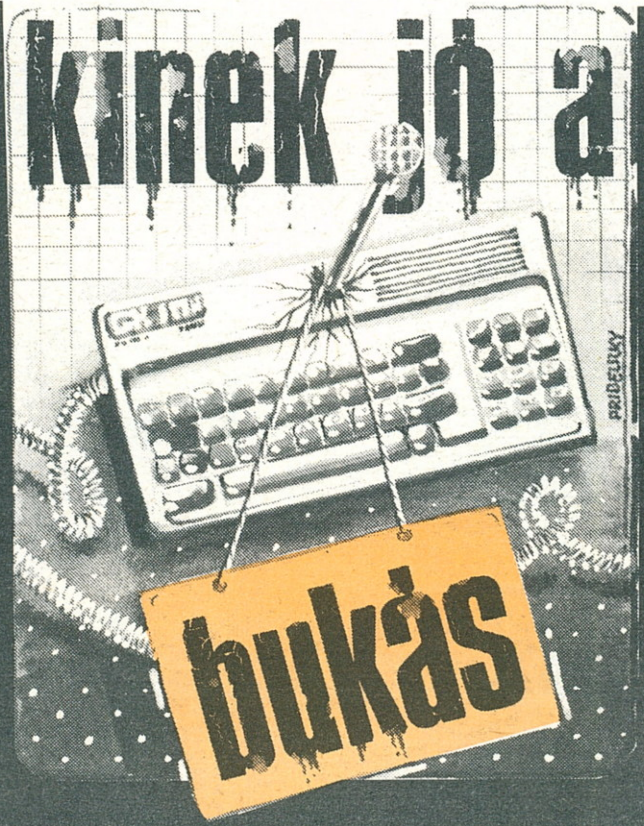
BIT-LET

ötlet SZÁMÍTÁSTECHNIKAI MELLÉKLET · 1986 NOVEMBER 38

kinek jó a bukás



Örömmel tapasztaltam, hogy vannak még olyan törzsolvasói a BIT-LET-nek, akik emlékeznek a régebben leírt dolgokra is, s még arra is vállalkoznak, hogy régebbi viselt dolgait a szerkesztő szemére olvassák. Történt ugyanis, hogy a múlt havi BIT-LET megjelenése után fölhívott egy olvasó, és a szememre hányta októberi jegyzetem néhány sorát. Abban ugyanis abbéli örömömet fejeztem ki, hogy nem árt a hazai számítástechnika ügyének, hogy sok helyen megbukott a számítógépesítés, megbuktak azok az összetákolt adatnyilvántartások, amelyek kis gépen akartak hozzá nem illő mennyiségű adatot tárolni. A telefonáló emlékezett még arra a régebbi szerkesztői jegyzetre, amelyben éppen azért füstölögtem, hogy ahol ilyen értelmetlen módon kezdik el a gépesítést, ott nyilvánvalóan előbb-utóbb kudarcot vallanak, s e kudarc láttán azt fogják hinni, hogy magában a számítógépben van a hiba, s egyszer s mindenkorra kiábrándulnak a dologból, és sürgősen visszatérnek a papiroshoz, gondolván, hogy az mégis biztonságosabb és megbízhatóbb. Nos, a két gondolat valóban ellentmondásos. Szerencsére azonban az élet nem mindenben igazolt engem. Vannak vállalatok, intézmények, ahol az történt, amit annak idején „jósoltam'', s miután a gép megjelenése, a nem neki való feladat megoldásának képtelensége több zűrt okozott, mint eredményt, egy idő után valóban a zárt szekrénybe vagy az igazgató bácsi fiacskájának asztalára került a számítógép. A vállalatnál pedig elkönyvelték, hogy a számítástechnika csak zűrzavart okozhat.

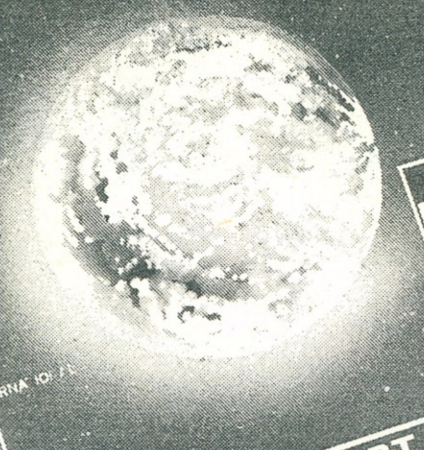
Más helyeken – s hál istennek ilyenből is van épp elég – azonban az történt, hogy a megkezdett munkát folytatták. Miután kiderült, hogy a gép nem bírja a feladatot végrehajtani, a szakembereknek sikerült a vezetőket meggyőzni arról, hogy a hiba nem a számítástechnikában van, hanem a gép és a feladat összhangjának a hiányában. Így történhetett, hogy a kicsinek bizonyult gép helyére nagyobb került, s az átmeneti zűrzavarok után a helyzet rendeződött, s a gép végre elkezdett dolgozni.

Hogy a kétféle következmény közül melyikből van több, nem tudom. Hogy a gépben csalódott vezetők, alkalmazottak mikor látják be majd tévedésüket, s mikor szavaznak újra bizalmat a számítógépnek, nem tudom. Egy azonban biztos. A kudarcoknak nagy hatásuk van. Aki egyszer kudarcot vallott valamivel, az legközelebb, ha hasonló dologhoz fog, ötször is átgondolja, hogy hogyan csinálja. Így hát valóban az a meggyőződésem, hogy a számítógépesítés kudarcai csak előre vihetik a számítógépesítés ügyét. Aki annak idején igazi szakértői támogatás nélkül, felelőtlenül fogott a dologhoz, az holnap, vagy talán ma, kétszer is meggondolja, hogy mit csináljon, és kinek a segítségével. S ez is hozzá tartozik a dologhoz, mármint a kivel. Mert nyilvánvaló, hogy igazi számítógépes szakember nem vállalt el olyan feladatot, amelyről már kezdetben látnia kellett volna, hogy megoldhatatlan. Hogy mégis voltak vállalkozók az ilyen munkákra, mindez azt bizonyítja csupán, hogy a számítógépes szakmában is voltak, vannak olyanok, akik pénzért a lelküket is eladnák. Márpedig tőlük megszabadulni a szakma presztízse érdekében is fontos. A kudarcok valószínűleg nem használtak az illetőknek. A nagy pénzekért elvégzett kis munkák, s ezek végeredménye talán hírbe hozta őket. Rossz hírbe. Bizakodhatunk benne, hogy helyettük a feladatot ugyanannál a cégnél legközelebb más kapja meg. Olyan, akinek nemcsak az a fontos, hogy mennyit kasszírozhat, hanem az is, hogy munkája eredményeként a számítógépet használóknak egy életre elmegy a kedvük a számítástechnikától, vagy éppen fordítva, egy életre elkötelezett hívei lesznek a számítógépnek, a számítógépesítésnek.

Angyalosi László

BELÜLRŐL

HÍROLDAL



TANULÓ VEZETŐ

A pennsylvaniai (USA) egyetemen egy Warp névre keresztelt szuperszámítógép kifejlesztésén munkálkodnak. A jelenleg elkészült példányt most tanítják meg az autóvezetésre. A számítógépet beszerelték egy gépkocsiba és érzékelő kamerákkal kötötték össze. Elérték, hogy a szupergép egy kilométeres sebességgel képes biztonságosan vezetni az autót. Továbbfejlesztési célkitűzés, hogy néhány hónap múlva mintegy ötven kilométeres sebesség mellett is idejében érzékelje az út, a forgalom, a jelzőtáblák változásait és megtegye a kellő lépéseket a biztonságos továbbhaladás érdekében.

ÖLTÖZKÖDÉS

A franciaországi Lectfa Systemes cég különleges, új módját gyakorolja a divatos ruhák kereskedelmének. Nem rendelkezik semmiféle készruha-raktárkészlettel. Butikjaiban a leendő vásárló katalógusokból kiválasztja a megfelelő ruhát, amit a próbafülkében egy képfeldolgozó és video tervezőrendszer a tükör előtt „ráilleszti". Míg a vevő meggyőződik választása helyességéről, a rendszer automatikus lézerletapogatással megállapítja a méreteket is. Az üzletekből ezután egy adatátviteli vonalon érkeznek a gyárba a ruhatípusokra, méretre, anyagra vonatkozó adatok és kezdődhet a gyártás.

JC LIPS

A 795 dolláros új számítógép az IBM PC-vel kompatibilis, sebessége 2–4-szeresen meghaladja az IBM PC sebességét. A gép a japán NEC cég V40-es processzorára épül. A rendszerkártya 256 Kbyte-os, RAM-ja 640 Kbyte-ig bővíthető. A rendszerbe tartozik egy beépített floppy egység (360 K), soros interface, párhuzamos printer interface billentyűzet, tápegység, színes vagy fekete-fehér monitor stb.

SZOFTVEREXPORT

Keresettek külföldön a Számítástechnika Alkalmazási Vállalat (SZÁMALK) számítógép-programjai, a vállalat gyorsan bővíti értékesítését a tőkés országokban. Legfontosabb piacának jelenleg Ausztria, az NSZK és Svájc számít. Legújabban japán üzletemberekkel kezdtek tárgyalásokat. A vállalat egyik legsikeresebb programtermékének az SZKI-val közösen kifejlesztett Softorg programcsalád bizonyult, melynek egyes moduljait többek között a Deutsche Bundesbanknál, a BMW-nél és a Schering AG-nél alkalmazzák. A SZÁMALK a világ legnagyobb számítógépgyártó cégének, az IBM-nek is adott el szoftvereket. A magyar vállalat szakemberei mikroszámítógép-programokon is dolgoznak. Az IBM PC-kre kifejlesztett speciálisprogramjaikat jelenleg több, NSZK-beli áruházi lánc is forgalmazza.

BÁV PC

A Bizományi Áruház Vállalat Budapest VIII., József körút 17. szám alatti üzlete teljes egészében a számítástechnikára szakosodott. Sokféle személyi számítógép kapható a boltban. Jelenleg az IBM kompatibilis PC-knek van nagy sikere. Az üzlet vezetése egy speciális szolgáltatással is kedveskedik a vásárlóknak. Minden csütörtökön 17 és 20 óra között PC-felhasználói klubot működtetnek.

-CLN

CHIPSZÁMOK

A világon egyre több helyen és egyre nagyobb számban gyártanak félvezető áramköröket, chipeket. Az egész világ félvezetőforgalma ma mintegy évi huszonötmilliárd dollár értékű. Ebből mintegy kilenc-kilenc milliárddal részesedik az amerikai és a japán chipipar. A legnagyobbak mellett több kisebb ország is konkurenssé válhat. Ilyen például Dél-Korea, ahol a kettőszázötvenhat K-s memóriachipeket a japánoknál is jóval olcsóbban, mindössze három és fél dollárért ajánlják, és így hamarosan óriási volumenben gyárthatják.

AUTÓRUGÓZÁS

A gépkocsik hagyományos kerékfelfüggesztésénél két tényező hat egymással szemben: ha elég lágy a felfüggesztés, akkor az útfelület változásait jól kiegyenlíti és simán, rázásmentesen halad az autó, viszont ilyenkor a kanyarokban válik bizonytalanná. Ennek kiküszöbölésére az angol Lotus cég a hagyományos rugózást olyan kísérleti rendszerrel helyettesítette, amely folyamatosan figyelemmel kíséri a kerekek mozgását és a pillanatnyi helyzetnek megfelelően, számítógép segítségével módosítja a felfüggesztést. Így a gépkocsi, a motorkerékpárokhoz hasonlóan a kanyarokban bedönthető, nagy gyorsításnál pedig előre dönthető. Az angol cég néhány autógyár – General Motors, Chrysler, Volvo – egyes prototípusain már be is mutatta új, számítógépes felfüggesztési rendszerét.

HOMELAB–4

Mint tudott, a Homelab–4 elnevezés, a dombóvári Color Ipari Szövetkezet által gyártott személyi számítógép az iskolaszámítógép pályázaton alul maradt. A szövetkezet vezetői azonban nem adták fel és nem állították le a gyártást. Keresik a megoldást gyártmányuk gazdaságos értékesítésére. Kooperációs tárgyalást kezdtek a Metriponddal, ahol mérlegek adatrgözítőjeként alkalmazzák majd a Homelab–4-eket. Ugyancsak kooperációra törekednek az ÉGSZI-vel és a Kecskeméti Automatizálási Főiskolával.

286 TURBO

Az egyesült államokbeli Western Computer cég újdonsága a Western Computer 286 Turbo elnevezésű számítógép. Az IBM PC/AT kompatibilis gép 512 Kbyte-os RAM-mal rendelkezik. A főkártyán elhelyezett rendszermemóriája 1 Mbyte-os. Háttértárolói opcionálisan 20–140 Mbyte-os lemezegységek, és 20–60 Mbyte-os szalagegységek lehetnek. A 286 Turbo többek között jól felhasználható szövegszerkesztésre, CAD/CAM feladatokra stb.



ÚJ!

SVÉD SIKKASZTÓ

Kiemelkedően nagyösszegű számítógépes csalást fedezett fel a svéd rendőrség. Egy harmincöt éves férfi tízmillió dollárnak megfelelő svéd koronát sikkasztott számítógép segítségével az ország legnagyobb nyugdíjpénztárából. A pénztár egyik számítástechnikai alkalmazottja a mágnes szalagon rögzített kamatokat nem a nyugdíjpénztár, hanem saját bankszámlájára utaltatta át.

KÖZVETÍTŐ

Színészek közvetítésére alakult számítógépes, információs cég az Egyesült Államokban. Az alapítás ötlete onnan származik, hogy számtalan esetben van szükség olyan színészekre, akik a színjátszás mellett valami különleges dologhoz is értenek. Ilyen igény lehet például, hogy az illető színész tudjon bokszolni vagy repülőgépből ejtőernyővel kiugrani. Az ilyen és hasonló adatokat gyűjtötték össze pillanatnyilag háromezerötszáz amerikai színészről.

ÜVEGSZÁL

Mint ismeretes az üvegszál az információ továbbítás igen hatékony eszköze. Most azonban az amerikai IBM cég jóvoltából jelentős előrelépés történt: megszületett az első üvegszálakból megépített chip. A kis méretű elektronikai eszköz (4,7x4,7 mm) az eddigi megoldásoknál négyszer gyorsabban alakítja át a fényt villamos jelekké és vissza. Az új chip többek között fotódetektort, félvezető lézert, erősítő részt és órajel-generátort tartalmaz. Segítségével üvegszálvezetőkön keresztül négyszáz millió Baud adatátviteli sebesség is elérhető.

MEZŐGAZDASÁG

Új szolgáltatásként számítógépes műtrágyázási programot készít taggazdaságainak a rákóczifalvi gabona- és ipari növények termelési rendszer, valamint a Szolnok Megyei Növényvédelmi és Agrokémiai Állomás. Az adatok számítógépbe táplálásához a táblánként vett talajminta laboratóriumi elemzésével megállapítják a talaj összetételét, jellemzőit. Számítógépbe kerül, hogy a területen korábban milyen növénykultúrákat termesztettek, mikor részesült a talaj szervestrágyázásban, hogyan alakult az öntözés. Mindezeket a számítógép feldolgozza és közli, hogy melyik táblán milyen termésátlag tervezhető, és optimális esetben milyen eredmény várható.

ŐSI SZÓ

Francia kutatók számítógépet hívtak segítségül egy különös vizsgálat elvégzéséhez. Arra voltak kíváncsiak, vajon a történelem előtti időkben, például a 450 ezer évvel ezelőtt élt őseink képesek voltak-e tagolt beszédre, a magán- és mássalhangzók, a szótagok és szavak kiejtésére. Behatóan tanulmányozták az ősember koponyáját, majd számítógép igénybevételével megrajzolták annak szájmodelljét. Megállapították, hogy az ősember képes volt a tagolt beszédre és feltehetőleg gazdag „szókinccsel" rendelkezett.

NÖVÉNYVÉDELEM

Eredményesen működik Pécsett az ország legnagyobb növényvédelmi adatbankja. Számítógépének memóriaegységében hatmillió adatot őriznek a baranyai szántóföldekről, s ezek alapján rövid idő alatt el tudják készíteni bármelyik gazdaság számára a vegyszeres gyomirtás leghatékonyabb, leggazdaságosabb programját. A ban is korszerű eljárást a pécsi Növényvédelmi és Agrokémiai Állomáson dolgozzák ki és a számítógépes növényvédelmet a megye nagyüzemi szántóterületének már 80 százalékára terjesztették ki.

COMPUPATIKUS

Szófiában számítógépre viszik az egyes gyógyszertárak keresett gyógyszereinek állományát. Eddig több mint harminc gyógyszertárat és többszáz gyógyszertípust vittek be a számítógépes rendszerbe. Ha az egyik gyógyszertárban nem kapják meg a keresett gyógyszert a betegek akkor nem kell találomra tovább menniük: ott helyben kapnak egy listát azon gyógyszertárakról, ahol a szükséges orvosság még kapható.

MEGA-PROJECT

Nagyszabású vállalkozásra fogott össze a két nagy nyugat-európai elektronikai cég a holland Philips és a nyugat-német Siemens. Mega Project elnevezéssel hárommilliárd márkás közös programot indítottak 1 és 4 Mbites memóriachipek fejlesztésére, és gyártására. Az 1 Mbitest a Philips, a 4 Mbitest a Siemens fogja gyártani. Egy 1 Mbites chip négyszer akkora információtömeg tárolására, másfélszeres sebességre és negyed akkora áramfogyasztásra képes, mint a ma legjobb 256 Kbites chip.

Utazás számítógéppel



Úgy gondoljuk, hogy a pályázatban részt nem vett olvasóink számára is hasznos, tanulságos a következő értékelés és a közölt program áttekintése.

A feladat lényege a következő volt:
Készítsünk programot, mely a városi (elsősorban pesti) tömegközlekedésben segít nekünk, azaz „megmondja", hogy valahonnan (pl. lakás) valahová (pl. munkahely) hogy tudunk a leggyorsabban eljutni (persze átlagosan!).
Rajzoljuk le a lehetséges útvonalakat az ábrán látható módon, s a vonalakra írjuk rá, hogy melyik közlekedési eszközt jelképezi, az átlagosan hány percenként jár, s a következő pontig mennyi az átlagos menetidője. Olyan programot kérünk, melybe ezt az ábrát valahogyan be lehet táplálni, s kiszámítja, hogy melyik útvonalon a legérdemesebb mennünk, hogy érünk oda leghamarabb.
Az ábrát és a további feltételeket a BIT-LET februári számában találhatják. (Ötlet – 1986. február 27.)
Úgy tűnik, ez a feladat igen nehéz volt pályázóinknak, illetve valószínűleg a megírására adott idő volt kevés. Ugyanis rengeteg lényegében jó, de nagyon barátságtalan és sok finomításra szoruló programot kaptunk. ***A fő nehézségek a következők voltak:***

1. Az adatok tárolására találjunk ki megfelelő struktúrákat.
2. Legyen megfelelő adatbevitel, javítás.
3. A feladatot alakítsuk át egyszerű legrövidebb út keresésére.
4. Be kellett építeni egy megfelelő, elég gyorsan működő, legrövidebb utat kereső algoritmust.
5. Próbáljunk javaslatot adni konkrét helyzetekben is.
6. A végeredményt megfelelő módon írjuk ki!

Olyan program nem érkezett, mely minden nehézséget sikeresen leküzdött volna. Az 1. és a 3. számút általában mindenki jól elintézte, talán a legegyszerűbb átalakítás az, hogy ha egy busszal lehet menni A-ból B-be, és tovább is B-ből C-be, akkor a gép generál egy közvetlen A–C utat a megfelelő adatokkal. (Vigyázni kell a szembe jövő buszra történő átszállásoknál!) Ezután minden élre ráírjuk a követési időköz felének és a menetidőnek az összegét, s kereshetjük a legrövidebb utat.
Voltak többen (pl. az alább közölt program írója is), akik az átszállásokat vették külön élnek, tehát növelték a csomópontok számát, s beiktattak közéjük olyan éleket, melyek csak átszállást jelképeznek. Ez a megoldás is jó, de a gépnek sok munkát ad, a gráfunk többszörösére növekszik!
A 4. nehézséget is igen sokan leküzdötték, de sajnos senki nem írta le pontosan, hogy milyen algoritmussal dolgozik. Néhányan legalább név vagy forráshely szerint hivatkoztak az algoritmusra, így pl. Salamon Csaba, az alábbi program írója D. Alcock könyvéből vette az algoritmust. Mivel ez a könyv valószínűleg mindenki számára hozzáférhető, így mi inkább egy másik lehetséges algoritmust írunk le, azt is új feladatként az utolsó oldalon.
Attól a néhány pályázótól eltekintve, akik úgy leegyszerűsítették a feladatot, hogy már használhatatlan és értelmetlen lett, illetve akiknek a programja rosszul futott; a legnagyobb nehézséget az adatbevitel, javítás okozta. Kevesen tudtak olyan beviteli módot találni, amellyel ezt a sok adatot gyorsan és kényelmesen be lehet vinni. Azonban szinte senki nem csinálta meg azt, hogy bármikor (adatbevitel után, legrövidebb út kiírása után) megfelelő formában megnézhessük az általunk beírt adatokat, s azokon kényelmesen javítgathassunk is. Enélkül a megnéző-javító rész nélkül a különben hasznos „adatok magnóra" című részeknek sem látjuk sok értelmét.
Az 5. nehézségről annyit, hogy több pályázónk megpróbálta legyőzni ezt is – több-kevesebb sikerrel.

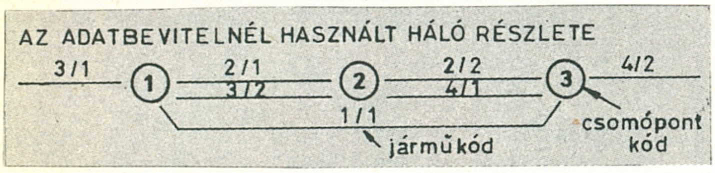
A 6. nehézség valószínűleg csak az idő rövidsége miatt volt az, mint utaltunk rá a programok barátságtalanok, a javaslatot sokan időadatok nélkül írták ki, többen nem engedik, hogy a csomópontokat elnevezzük, vagy hogy az utakra ráírjuk a járművet – el lehet képzelni, hogy milyen lesz ezek után a végeredményközlés!

S most lássuk az általunk legjobbnak érzett programot, melyet Salamon Csaba, tiszakécskei pályázónk írt C 64-re.

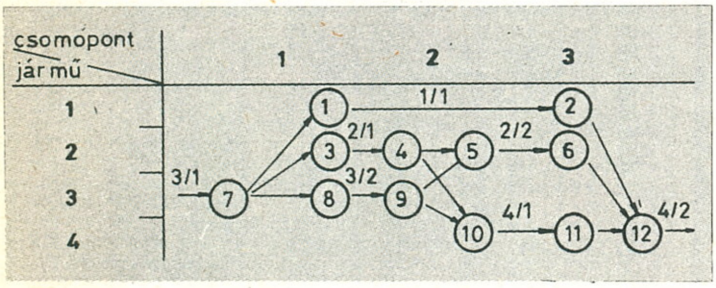
A program hálószámító algoritmusa Donald Alcock: Ismerd meg a BASIC nyelvet c. művében a 106. oldalon leírt legrövidebb út számítása szerint működik. A legkedvezőbb használhatóság érdekében a START- és a CÉL-állomások fel vannak cserélve, így az A tömb 2. oszlopa az adott csomópont céltól való legrövidebb távolságát (idő) tartalmazza. Az ott leírt program azonban nem alkalmas arra, hogy egy reláción belül több jármű közlekedését is figyelembe vegye, valamint a csomópontokon történő átszállások érzékelésére sem alkalmas ebben a formában.

Ezért szükséges a hálószámítás alapjául létrehozni egy bővített hálót a következők szerint.

Az adatbevitelnél használt háló részlete



A bővített háló ennek megfelelően a következő lesz



A bővített háló létrehozásáról a felhasználó nem szerez tudomást, az adatbevitelnél és az eredmények kiíratásakor is az eredeti háló jelölései az érvényesek.

Korlátozások: az eredeti hálóra vonatkoznak.

1. Egy jármű egy csomópontba legfeljebb egy irányból érkezhet, és egy irányba indulhat tovább.

2. Olyan csomópont, amelyben csak beérkező vagy csak továbbhaladó jármű van értelmezve, a start- és a célállomások kivételével nem lehet. ($\Sigma be * \Sigma ki \neq 0$)

3. A csomópontok, valamint a járművek számára vonatkozóan a következő korlátozásnak kell érvényesnek lenni

CM – csomópontok száma

JM – járművek száma

PC(i) – az i-edik csomópontba beérkező és az onnan továbbhaladó járművek számának szorzata
$\Sigma be(i) * \Sigma ki(i) = PC(i)$

SC(i) – az i-edik csomópontba beérkező és az onnan továbbhaladó járművek összege

$$JM*25+CM*20+JM*CM*9+$$
$$23\sum_{i=1}^{CM} SC(i)+6\sum_{i=1}^{CM} PC(i)<32000$$

Jelenleg $CM<=30$
$JM<=20$
$PC(i)_{max}<=25$
$SC(i)_{max}<=10$

CM a start- és a célállomással együtt, JM a gyaloglással együtt értendő.

A korlátozások a 80, 90, 450 sorok átírásával a fenti összefüggés figyelembevételével változtathatók meg.

4. A követési időkre és közvetlen utakra $0<t<=60$.

A program előbb a CM és a JM értékeket kéri, majd az egyes járművek nevét – JN$(JM) – és követési idejét – JK(JM) – valamint a csomópontok neveit – CS$(CM) –.

Ezután a közvetlen utakat kéri járművenként, tehát először be kell adni az összes olyan utat, ahol gyalog megyünk, s így tovább. A képernyő jobb oldalán fel vannak sorolva a sorszámozott csomópontok, így egy út bevitelénél a mettől és meddig kérdésekre csak a megfelelő csomópont sorszámával kell válaszolni.

Újabb járműre a „←" billentyűvel lehet átlépni.

A közvetlen utakra vonatkozó adatokat a CS%(1, JM, CM), illetve a B1(JM, CM) tömbökben helyezi el.

Az adatbevitel közben ellenőrzi a PC(i) és az SC(i) értékeket. 0 idejű utat nem engedélyez. Egy járműnek ugyanabba a csomópontba beérkezését, illetve ki- vagy továbbindulását csak egyszer engedélyezi.

Az adatbevitel után ellenőrzi a hálót szakadási hely szempontjából, majd kialakítja a bővített háló A(N, 3) és B%(E, 2) tömbjeit.

N – a bővített háló csomópontjainak a száma

E – a bővített háló útjainak száma

Ezután meghatározza az optimális utat, valamint a bővített háló csomópontjainak legkisebb távolságát (idő) a célállomástól (várakozási idő=JK(I)/2).

Végezetül kiírja a legjobb utat „járművel a csomópontig" alakban.

Ezután át kell térni az egyes csomópontokban követendő stratégiát segítő részre.

A következő információkat kell megadnunk:

1. Melyik csomópontban vagyunk? (Itt is fel vannak sorolva a csomópontok, csak a megfelelő csomópont sorszámát kell beírni.)

A célállomás kódját beírva a futás befejeződik.

2. Mennyi ideje tartózkodunk itt? (TA)

Az átlagos várakozási idők módosításához szükséges

$$JA(I) = \frac{JK(I)-TA}{2}$$

Ha $JA(I)<0 \Rightarrow JA(I)=0$

3. Mely járművek tartózkodnak a csomópontban?

Az itt megadott sorszámoknak (itt ugyanis a járművek felsorolása látható a képernyőn, s abból kell sorszám beírásával választani) megfelelő járművek várakozási ideje nullázódik: JA(I)=0.

Ha erre a kérdésre a gyaloglás kódját, azaz 0-t adunk, a program kiírja az adott csomópontban a feltételeknek megfelelően valamennyi e csomópontból továbbhaladó jármű esetén a célállomás eléréséig várhatóan szükséges időt.

Tetszőleges gomb nyomására az aktuális csomópont kérdéshez tér vissza.

A program kilépés előtt lehetőséget nyújt CS$(CM) JN$(JM), JK(JM), CS%(1, JM, CM) és B1(JM, CM) tömbök mentésére.

Meglevő állomány esetén a magnó állomány elejére állítása után RUN 3800-zal indítható.

(Szín nincs, hang van, RUN/STOP nincs letiltva.)

Fontosabb modulok:

0–100	– szabályok, korlátok kiírása CM, JM bekérése
100–180	– járművek nevei és járművek követési idejei JN$(JM), JK(JM)
200–280	– csomópontok nevei CS$(CM)
300–500	– közvetlen utak utazási idejének bevitele (CS%(1, JM, CM); B1(JM, CM))

ötlet

U t a z á S
SZÁMÍTÓGÉPPEL



```
10 PRINT""TAB(15)"KORLATOZASOK"
20 PRINT"START ES CEL ALLOMASOKKAL EG
YUTT"TAB(52)"MAX CSOMOPONT"
30 PRINT"A RELACIOKBAN SZEREPLO JARMU
VEK SZAMA"TAB(52)"MAX DB"
40 PRINT"AZ EGY CSOMOPONTRA ERKEZO E
S A"
50 PRINT"CSOMOPONTROL INDULO JARMUVEK S
ZAMANAK"
60 PRINT"   SZORZATA MAX "25
62 PRINT"AZ EGY CSOMOPONTRA ERKEZO E
S A"
65 PRINT"CSOMOPONTROL INDULO JARMUVEK S
ZAMANAK"
70 PRINT"    OSSZEGE MAX "10:QL=2:MI=3
80 PRINT"CSOMOPONTOK SZAMA:";:QF=12:
MA=30: GOSUB,10000:PRINT:IFQF=0THEN10
90 CM=Q%-1:PRINT"JARMUVEK SZAMA   :";
:QF=12:MA=20:GOSUB 10000:IFQF=0THEN10
95 JM=Q%-1:DIM CS%(1,JM,CM),B1(JM,CM),CS
$(CM),JM$(JM),JK(JM),JA(JM)
100 JM=Q%-1:PRINT:PRINT"   FOLYTA
TAS TETSZ. BILLENTYURE"
105 GETA$:IFA$=""THEN105
110 REM ***JARMUVEK ***
120 PRINT""TAB(12)"  JARMUVEK  "
130 PRINT"SORSZAM    JARMU NEV      KOV
ETESI IDO"
135 FORI=1TO39:PRINT"-";:NEXT:PRINT
136 JM$(0)="GYALOG":PRINT0TAB(8)JM$(0);
140 PRINTTAB(28)JK(0):FORJ=1TOJM:PRINTJT
AB(8);:QF=13:QL=15:GOSUB10000
145 IFQF=0THEN110
150 JM$(J)=Q$:PRINTTAB(29);:QF=11:QL=4:M
I=0:MA=60:GOSUB10000:IFQF=0THENGOTO110
160 JK(J)=Q:PRINT
165 NEXT
170 POKE782,5:POKE781,24:SYS65520:PRINT"

180 GETA$:IFA$=""THEN180
200 REM ***CSOMOPONTOK***
210 PRINT""TAB(12)"  CSOMOPONTOK  "
220 PRINT"SORSZAM   CSOMOPONT NEV"
225 FORI=1TO40:PRINT"-";:NEXT
230 PRINT0,"START":CS$(0)="START"
240 FOR C=1TO CM-1:PRINTC,:QF=13:QL=20:G
OSUB10000:IFQF=0THEN2000
250 CS$(C)=Q$:IF C>19THENPRINT"":PRINT"
                                       "
260 PRINT:NEXT:PRINTCM,"CEL"
270 POKE782,5:POKE781,24:SYS65520:PRINT"

280 CS$(CM)="CEL":GETA$:IFA$=""THEN280
300 REM***KOZVETLEN UTAK***
305 CS%(1,0,CM)=1:N=2
310 FORJ=0TOJM:S=0:PRINT""TAB(10)"KOZ
VETLEN UTAK FELVITELE":PRINT"
316 GOSUB5000:POKE782,10:POKE781,24:SYS6
5520:PRINT"(<←>=KOVETKEZO JARMU)";
320 PRINT"CS.P.CS.P.MENET":PRINT"

330 IFS>14 THENPRINT"                  "
333 S=S+1:PRINT"  ";:MI=0:MA=CM:QF=12:QL
=2:GOSUB10000:IFQF=0THEN500
340 CK=Q%:PRINTTAB(4)"- ";:QF=12:GOSUB10
000:IFQF=0THEN500
350 CV=Q%:PRINTTAB(10);:QF=11:QL=4:MA=60
:GOSUB10000:IFQF=0THEN500
360 UT=Q:REM***ELLENORZES****
370 IFCS%(1,J,CK)<>0THEN480
380 IFCS%(0,J,CV)<>0THEN480
390 IFCV=CKORUT=0THEN480
400 K1=1:K2=0:V1=0:V2=1:FORJA=0TOJM
410 IFCS%(1,JA,CK)<>0THENK1=K1+1
420 IFCS%(0,JA,CK)<>0THENK2=K2+1
430 IFCS%(1,JA,CV)<>0THENV1=V1+1
440 IFCS%(0,JA,CV)<>0THENV2=V2+1
450 NEXTJA :IFK1*K2>25ORV1*V2>25ORK1+K2>
10ORV1+V2>10THEN480
460 CS%(1,J,CK)=N:N=N+1:CS%(0,J,CV)=N:N=
N+1:B1(J,CV)=CK*100+UT:PRINT: GOTO330
480 PRINT:PRINT"HIBAS ADAT  "
490 GOTO330
500 NEXT:CS%(0,0,0)=0
510 REM ***ELLENORZES.A,B TOMB MERETE**
520 REM N=CSOMOPONTOK SZAMA A(N,3)
530 REM E=UTAK SZAMA         B(E,2)
532 REM S=CEL CSOMOPONT
535 REM H=KEZDO CSOMOPONT
540 REM ******************************
545 PRINT""TAB(16)"ELLENORZES"
550 E=0:FORJ=0TOJM:FORC=0TOCM:IFB1(J,C)<
>0THENE=E+1
560 NEXTC,J
570 FORC=0TOCM:K1=0:V1=0:FORJ=0TOJM
580 IFCS%(0,J,C)<>0THENV1=V1+1
590 IFCS%(1,J,C)<>0THENK1=K1+1
600 NEXTJ:P1=V1*K1:IFP1=0THENPRINT"
hibas csomoPont.Kezdd elolrol":end
610 E=E+P1:NEXTC:DIMA(N,3),B%(E,2)
620 REM ***A TOMB ELOKESZITES****
630 FOR I=1TON:A(I,1)=1:A(I,2)=10000:A(I
,3)=0:A(I,0)=0:NEXT:A(1,2)=0
640 S=1:H=N:REM***KEZD.ES VEGPONTOK***
700 REM***KOZVETLEN UTAK***
705 PRINT""TAB(3)"ELOKESZITES.A
 ES B TOMBOK TOLTESE"
710 P=0:FORJ=0TOJM:FORC=CMTO0STEP-1
720 IF B1(J,C)=0THEN750
730 C1=INT(B1(J,C)/100):A=CS%(0,J,C):B=C
S%(1,J,C1):R=B1(J,C)-100*C1
740 P=P+1:B%(P,2)=B:B%(P,0)=INT(100*R+.5
):B%(P,1)=A(A,0):A(A,0)=P
750 NEXTC,J
800 REM***ATSZALLASOK LATSZAT UTAK***
810 FORJ=0TOJM:FORC=CMTO0STEP-1
820 A=CS%(1,J,C):IFA=0THEN880
830 FORJB=0TOJM:R=JK(J)/2:B=CS%(0,JB,C)
840 IFB=0THEN870
850 IFJ=JBTHENR=0
860 P=P+1:B%(P,2)=B:B%(P,0)=INT(100*R+.5
):B%(P,1)=A(A,0):A(A,0)=P
870 NEXTJB
880 NEXTC,J
2000 REM***HALOSZAMOLO***
2100 PRINT""TAB(14)"HALOSZAMITA
S"
2230 I=S:GOTO2320
2250 I=I+1:IFI<=NTHEN2300
2270 I=1:K=K+1:IFK>1THEN2440
2300 IFA(I,1)=0THEN2250
2320 J=A(I,0):IFJ=0THEN2420
2330 K=0
2340 T=A(I,2)+B%(J,0)/100
2350 L=B%(J,2):IF A(L,2)<T THEN2400
2370 A(L,1)=1:A(L,2)=T:A(L,3)=I
2400 J=B%(J,1):IF J<>0THEN2340
2420 A(I,1)=0:GOTO2250
2440 REM ***ELOKESZITES VEGE***
2500 X=A(H,3):CA=1:GOTO2515
2510 POKE782,5:POKE781,24:SYS65520:PRINT
"FOLYTATAS TETSZ. BILLENTYURE"
2512 GETA$:IFA$=""THEN2512
2515 S=0:PRINT""TAB(15)"LEGJOBB UT"
2520 PRINT" JARMU"TAB(18)" CSOMOPONTIG"
2530 FORI=0TO38:PRINT"-";:NEXT:PRINT
2540 S=S+1:IFS=15THEN2510
2550 IFCA=CMTHEN2590
2560 I=1:GOSUB4000:PRINTJM$(JA);:X=A(CS%
(1,JA,CA),3)
2570 I=0:GOSUB4000:PRINTTAB(18)CS$(CA)
2580 X=A(CS%(0,JA CA),3):GOTO2540
2590 POKE782,5:POKE781,24:SYS65520:PRINT
"FOLYTATAS TETSZ. BILLENTYURE"
2595 GETA$:IFA$=""THEN2595
3000 REM ***TETSZ.CS.P.OPT.STRATEGIA***
3010 REM CA=AKTUALIS CS.P.
3020 REM JA()=CS.P.KI JARMUVEK VAR.IDO
3030 CI$=" OPTIMALIS STRATEGIA TETSZ.
CSOMOPONTBAN"
3040 PRINTCI$:GOSUB5000
3045 POKE782,5:POKE781,24:SYS65520:PRINT
""CM"=KERDEZES VEGE";
3050 PRINT"AKTUALIS CS.P.?"TAB(40);
:QF=12:MI=0:MA=CM:QL=2:GOSUB 10000
3060 IFQF=0THEN3000
3070 IFQ%=CM THEN3650
3080 CA=Q%:PRINT:PRINT"";CS$(CA):PRINT"

3090 QL=4:GOSUB10000:IF QF=0THEN3000
3100 TA=Q:FORJ=0TOJM:IFCS%(1,J,CA)=0THEN
JA(J)=10000:GOTO3120
3110 JA(J)=(JK(J)-TA)/2
3120 IF JA(J)<0THENJA(J)=0
3130 NEXT
3200 PRINTCI$
3210 PRINT""TAB(20)"JARMUVEKT":PRIN
TTAB(19);:FORI=1TO16:PRINT"-";:NEXT
3220 PRINT:FORJ=0TOJM:PRINTTAB(19)""JTA
B(23)""JM$(J):NEXT
3225 POKE782,5:POKE781,24:SYS65520:PRINT
"D=ADATBEVITEL VEGE";
3230 PRINT"CSOMOPONTBAN"TAB(40)"TA
RTOZKODO":PRINT"
```

```
3240 QF=12:GOSUB10000:IFQF=0THEN3200
3250 JA=Q%:IFJA=0THEN3500
3255 IFCS%(1,JA,CA)=0THENPRINT"■■ HIBAS
ADAT":GOTO3240
3260 JA(JA)=0:PRINT"■■ ";JM$(JA):GOTO324
0
3500 PRINTCI$"■"
3505 PRINTTAB(10)"■"CS$(CA):PRINTTAB(5)"
jarmu"tab(20)"utazasi ido a celig"
3510 FORI=1TO39:PRINT"-";:NEXT:PRINT
3520 FORJ=0TOJM:IFCS%(1,J,CA)=0THEN3600
3530 PRINTJM$(J)TAB(20)A(CS%(1,J,CA),2)+
JA(J)TAB(30)"PERC"
3600 NEXT
3610 POKE782,5:POKE781,24:SYS65520:PRINT
"■FOLYTATAS TETSZ. BILLENTYURE■"
3620 GETA$:IFA$=""THEN3620
3630 GOTO3000
3640 PRINT"■"
3650 PRINT"■■■■■■■■■■■■HA MENTENI AKAR
SZ IRD F1":FORI=1TO300:NEXT:PRINT""
3655 FORI=1TO100:NEXT:GETA$:IF A$=""THEN
3650
3660 IF A$<>"■"THENEND
3700 REM MENTES
3710 OPEN 1,1,1,"1"
3720 PRINT#1,CM
3730 PRINT#1,JM
3740 FORI=0TO1:FORJ=0TOJM:FORC=0TOCM:PRI
NT#1,CS%(I,J,C):NEXTC,J,I
3750 FORJ=0TOJM:FORC=0TOCM:PRINT#1,B1(J,
C):NEXTC,J
3760 FORJ=0TOJM:PRINT#1,JM$(J):NEXTJ
3765 FORJ=0TOJM:PRINT#1,JK(J):NEXTJ
3770 FORC=0TOCM:PRINT#1,CS$(C):NEXTC
3780 CLOSE 1:END
3800 REM TOLTES
3810 OPEN 1,1,0,"1"
3820 INPUT#1,CM
3830 INPUT#1,JM
3835 DIM CS%(1,JM,CM),B1(JM,CM),CS$(CM),
JM$(JM),JK(JM),JA(JM)
3840 FORI=0TO1:FORJ=0TOJM:FORC=0TOCM:INP
UT#1,CS%(I,J,C):NEXTC,J,I
3850 FORJ=0TOJM:FORC=0TOCM:INPUT#1,B1(J,
C):NEXTC,J
3860 FORJ=0TOJM:INPUT#1,JM$(J):NEXTJ
3865 FORJ=0TOJM:INPUT#1,JK(J):NEXTJ
3870 FORC=0TOCM:INPUT#1,CS$(C):NEXTC
3880 CLOSE 1:N=CS%(0,0,0):GOTO510
4000 REM CSOMOPONT ES JARMU MEGTALALO
4010 REM BE***
4020 REM X  -BOVITETT HALO AKTUALIS CSP
4040 REM I  -ERKEZES I=0,TAVOZAS I=1
4050 REM KI***
4060 REM CA -AKTUALIS CSOMOPONT
4070 REM JA -AKTUALIS JARMU
4100 FOR CA=0TOCM:FORJA=0TOJM
4200 IFCS%(I,JA,CA)<>XTHENNEXTJA,CA
4300 RETURN
5000 REM***CSOMOPONT KIIRO"
5100 PRINT"■■■■"TAB(20)"■CSOMOPONTOK":PR
INTTAB(20);:FORI=1TO19:PRINT"-";:NEXT
5200 PRINT:FORC=0TOCM:PRINTTAB(15)"■"CTA
B(19)"■"CS$(C):IFC=17 THENC=CM-1
5300 NEXTC:RETURN
10000 REM**INPUT USING**
10060 POKE54273,68:POKE54278,240:POKE542
96,15:POKE54277,15:REM *HANG ELOKESZ
10070 QF=QF-10:IFQF<9THENQB=0:Q$="":Q=0:
GOTO10095
10075 IFQF<3THENQ=Q%
10080 QF=QF-10:IFQF<3THENQ$=RIGHT$(STR$(
Q),LEN(STR$(Q))-1)
10090 QB=LEN(Q$):PRINTQ$;
10095 IFQB=0THENPRINT"?■";
10100 IF QB<>QLTHENPOKE204,0
10105 GETQ1$:IFQ1$=""THEN10100
10110 POKE204,1:PRINT" ■";:IFQ1$=CHR$(13
)THENGOTO10200
10120 IFQ1$=CHR$(20)ANDQB<>0THENQB=QB-1:
Q$=LEFT$(Q$,QB):PRINT"■■";:GOTO10095
10125 IFQ1$="←"THENQF=0:RETURN
10130 IFQ1$<CHR$(32)ORQ1$>CHR$(94)ORQ1$=
CHR$(34)ORQ1$=","ORQ1$=":"THEN10300
10135 IFQF=2AND(Q1$>CHR$(57)ORQ1$<CHR$(4
8))THEN10300
10137 IFQF=1AND(Q1$>CHR$(57)ORQ1$<CHR$(4
6)ORQ1$="/")THEN10300
10140 IF QB<QLTHENQB=QB+1:GOTO10160
10145 GOTO10300
10160 Q$=Q$+Q1$:PRINTQ1$;:GOTO10095
10200 IFQF=3ANDQ$>"!"THEN10220
10202 IFQF<3THENQ=VAL(Q$)
10205 IFQF=2ANDQ>32500THEN10300
10210 IFQF=3OR(Q<MIORQ>MA)THEN10300
10215 IFQF=2THENQ%=INT(Q)
10220 POKE204,1:RETURN
10300 POKE54276,17:FORI=1TO50:NEXT:POKE5
4276,16:GOTO10095
```

	– 360–490 – csomópontra vonatkozó korlátozások ellenőrzése
510–600	– felesleges csomópont v. szakadási hely szerinti ellenőrzés
630	– A(N, 3) tömb előkészítése
640	– cél-start csere
700–750	– A(N, 3) és B%(E, 2) tömbök töltése (közvetlen utak)
800–880	– A(N, 3) és B%(E, 2) tömbök töltése (átszállások, átutazások)
2000–2440	– hálószámítás (Alcock könyv alapján)
2500–2590	– legjobb út kiíratása
3000–3260	– optimális stratégia tetszőleges csomópontban: kezdeti feltételek bekérése és az optimális stratégiát segítő adatok számítása
3500–3630	– optimális stratégia kiíratása
3640–3780	– mentés, ill. kilépés
3800–3880	– visszatöltés
4000–4300	– csomópontkeresés (legjobb út kiíratásához)
5000–5300	– csomópont kódtáblázat kiíratás
10000–10300	– adatbekérő rutin

Adatbekérő rutin:

QF – funkció kód

QF MOD 10=1	REAL
QF MOD 10=2	INTEGER
QF MOD 10=3	STRING
INT (QF/10)=1	új adat
INT (QF/10)=2	régi adat kiírása az adatbekérés előtt

QL	– bekérendő karakterek
Q, Q%, Q$	– be-, ill. kimenő adatok
MI	– alsó határ
MA	– felső határ
QB	– már bevitt karakterek száma
Q1	– utolsónak bevitt karakter

Megjegyzések:

– Javítás csak DEL billentyűvel.
– Kurzor vezérlők letiltva.
– „←" billentyűvel lehet egy szinttel magasabbra, illetve továbblépni bizonyos esetekben.
– Ha nem lehet több karaktert bevinni, a kurzor nem villog.
– Adatbevitelt RETURN billentyűvel zárjuk.
– Használat közben tekintsünk néha a 25. sorban levő kezelési utasításokra.

ötlet

POSTA



A Posta rovatba illene az alábbi levél, de mert a válasz a szokásosnál hosszabbra és érdekesebbre sikeredett, úgy gondoltuk, hogy túlnőtte a rovat kereteit. Ezért lett belőle külön oldal.

Tisztelt Szerkesztőség!

Több kérdéssel fordulok önökhöz. Az egyik óriási problémám, hogy Commodore 64-es gépem van, s nem vagyok megelégedve a géphez kapható dokumentációval. Az egyetlen „jó" gépkönyv az Úry-féle kétkötetes könyv. Ezt, ára miatt egy kis szürke, polgári halandó, mint én, nem tudja megvenni a zsebpénzéből.

MITŐL ILYEN DRÁGÁK AZ EFFÉLE KÖNYVEK? MITŐL ILYEN CSAPNIVALÓAK?! Példának okáért: „Csodásan" be van írva a sprite-készítés. Pompás, remek! Csak azt az „egyet" (!) nem írja, hogy milyen értékeket kell bePOKE-olni Pl.: a 832-es címtől kezdve. Egy átlag sprite-nál ugye így valahogy: veszek egy 24*21-es téglalapot, megtervezem az ábrát, átszámítom az értékeket 2-es számrendszerbe, összeadom az egy sorban levőket (24/8=3 db ilyen van egy sorban), így 63 adatot kapok. Na de honnan tudná az a szerencsétlen gép, hogy én az ábrának melyik részét akarom pirosra festeni, és melyiket zöldre, vagy hupililára?! Ami a gépkönyvben szerepel, hogy a DATA-kba A, B, C betűket írok, (mint a SIMON'S-ban) az szerintem annyira működőképes, mint amennyire áttekinthető, megérthető. Ezért a pénzért igazán vehetné a szerző a fáradságot, hogy ne szaknyelven írjon, hanem némi fiatalságot beleszőve, spontán magyarul írjon! Nagy munka lenne? Szóval ez az 1. kérdésem.

A második ugyancsak sprite-tal kapcsolatos. Ha az ember elővesz egy Block-katalógust (ez, úgy tudom, nem szerepel a könyvben), hogy mégis milyen címtől töltse be az adatait, ránéz, és azt mondja, hogy atyaúristen! Fel nem fogom, hogy minek lehet ennyi? (Ezek közül csak az első 4-et használhatom szabadon, a többit BASIC kezdőcím eltolással.) Csak úgy volna értelme, ha a betöltött adatokat felváltva használhatom. De hogyan?! A POKE 53269, sprite száma 2-es számrendszerben szerintem nem használható. Akkor mi, hogyan? Sejtelmem sincs.

Az utolsó, immár nem annyira elmarasztaló problémám, hogy hogyan lehet hirdetni. Valami olyasmit olvastam, hogy aki beküldi az előfizetési csekket (vagy micsodát) az már dörzsölheti a tenyerét, s törheti a fejét, hogy mit is hirdessen. Hát ez kész kiszúrás. Minthogy még mindig a szürke polgárok értékekben szegény életét élem, csak a BIT-LET-es számokat veszem. (Ami csak ritkásan jelenik meg, mint a Loch Ness-i szörny.) Szóval, mint már említettem ez kitolás. (Tetszik látni, már remeg a kezem: a szívinfarktus első jelei.) Szóval, **mint C 64-tulajdonos, szívesen venném fel a kapcsolatot más, 172 magas, lehetőleg 20 év alatti fiatalokkal programcsere céljából.**

Szerintem ésszerűbb lenne, ha mint más kiadóknál, pénz ellenében lehetne hirdetni. Na nem olyan csillagászati összegekért, mint az EXPRESSZ-ben, csak amolyan „jelképes összegért".

Üdvözlettel **Bognár Ákos,** III. oszt. gimn. tan. 1124 Budapest, Levendula u. 20.

Leveleiben föltett kérdései között akadnak részünkről megválaszolhatatlanok, illetve olyanok, amelyekre legfeljebb véleményünket mondhatjuk el.

1. Ami a könyvek árát illeti, úgy tűnik, hogy ezeknek a könyveknek az árát a kereslet-kínálat törvényei szabályozzák. Ennyiért is elmennek. A furcsa csak az, hogy miért éppen egyedül a számítástechnikai könyvek ára alakul így?

2. Hogy melyik könyv csapnivaló és melyik nem, ebben nem biztos, hogy egyet értünk. Abban viszont igen, hogy jobb lenne, ha a könyvek minél nagyobb százaléka íródna a szakmai „bikkfanyelv" helyett közérthetően.

3. A hirdetésügyben ezúton közöljük önnel is és olvasóinkkal is, hogy apróhirdetéseiket cserebereügyekben, hasonló gépekkel foglalkozó társak keresése ügyében már régóta ingyen és bérmentve közöljük. Semmiféle előfizetői igazolás vagy efféle nem kell hozzá.

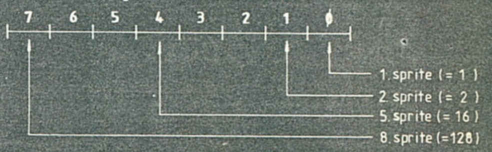
És végül, de nem utolsósorban az ígért hasznos információk.

Amit a sprite-kezeléshez még tudni kell:

1. A látható képernyő egy 320*200 darabból álló ponthalmaz. Ebből a sprite egy olyan látható figura, amelyet 24*21 db pontból állítunk össze, és lehetőségünk van arra, hogy csak a koordináták meghatározásával gyorsan (hardver úton) mozgassuk. A sprite koordinátáit azonban egy 512*256-os területen helyezhetjük el, így – arányosan – a képernyő bármelyik oldalán kívülre is koordinálhatjuk.

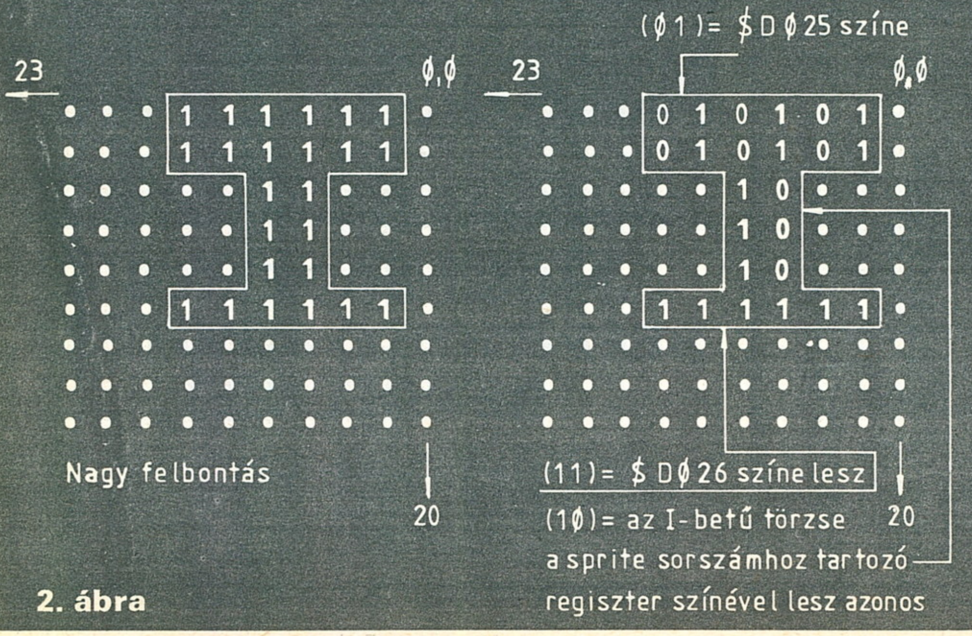
2. A sprite-okat nagyfelbontású üzemmódban vagy többszínű üzemmódban használhatjuk. A nagyfelbontású módot választva a 24*21-es pontterület minden pontját kihasználhatjuk, de többszínmódban ez a felére csökken (12*21-re) azáltal, hogy a figura egyetlen látható pontját a rendelkezésünkre álló területből KÉT ponttal határozzuk meg, s így színt is adhatunk neki a következőképpen:

A $D01C regiszternek a sprite-nak megfelelő bitjét 1-re kell állítani.



Ezzel a regiszterrel bekapcsoltuk a sprite többszínű üzemmódját. (Ne feledjük ilyenkor a figurális felbontás 12*21!)

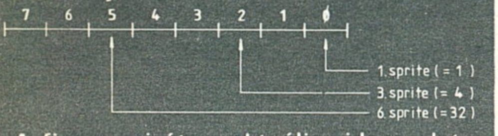
Ha most megnézzük a duplaszéles pontok jelentését, akkor máris színezni tudjuk a sprite-jainkat:



2. ábra

Ha a bitpár:
00 – akkor a pont színe egyezik a háttér színével
01 – akkor a pont a $D025(53285) regiszterbe vitt színértékkel egyezik meg
10 – akkor a $D027 – $D02E (53287–53294) regiszterek közül az lesz a színmeghatározó, amelyik a sprite sorszáma szerint következik. (első sprite=53287, második sprite=53288, stb.)
11 – akkor a $D026(53286)-os regiszterbe vitt szín értéke szerint lesz színezve.
Pl. az I-betű alakja nagyfelbontású és többszínű üzemmódban a 2. ábrán látható (Feltételezem a 24*21-es sprite tervezését és kiszámítását az olvasó már ismeri.)

3. A megtervezett és kiszínezett sprite-ot meg is kell jelenítenünk. Ehhez a $D015(53269) címnek a sprite-hoz tartozó bitjét 1-be kell állítani:



A figura rajzát az aktuális videoszeletből veszi a VIDEO-CHIP, mégpedig ahol a képernyőtár van, annak az 1 Kbyte-os területnek az utolsó 8 byte-jának az értékeit veszi alapul. (Megjegyzés: a videoszelet, a memóriából kiválasztott 16 Kbyte-os terület. Kijelölése a $DD00 és $DD02-es címekkel lehetséges, alapállapotban a 0-adik címtől veszi a memória legalsó 16 Kbyte-ját. Mivel a rendszer ezen a területen belül $400-ra helyezi a képernyőtár startcímét, így a hozzáadott 1 Kbyte – 8 eredménye $7F8 (decimálisan 2040) lesz. Ettől a címtől kezdve helyezhetjük el a sprite-ok rajzához tartozó startcímeket. Ez azért is lehetséges, mivel képernyőtárnak elegendő 40*25=1000 byte és így a teljes 1 Kbyte (=1024) nincs igazán kihasználva!)

A sprite-okhoz rendelt 8 regiszter értékeit a következőképpen állíthatjuk be arra a startcímre, ahova a sprite-ok rajzadatait ténylegesen is elhelyezhetjük:

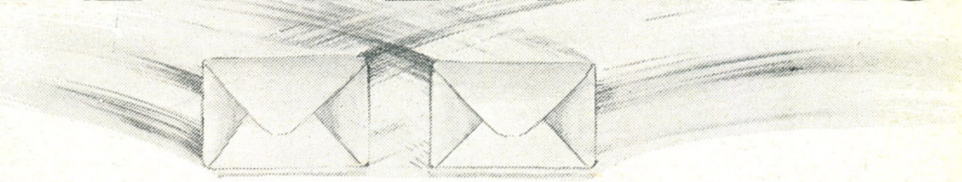
A 16 Kbyte=16*1024=16384 byte. Ha ezt elosztjuk 256-tal, akkor 64-et kapunk, azaz pont annyit, amennyi 1 sprite ábrázolásához elegendő. Ebből következik, hogy 256 sprite-rajzot tudunk elhelyezni, de ezt korlátozzák azok a tények, hogy csak 8 sprite-ot tudunk amúgy is csak megjeleníteni, továbbá a képernyőnek is kell levennünk ebből területet, a BASIC interpreternek is kell terület, sőt a sprite-okat mozgató és felhasználó programnak is kell terület, így egy olyan 64 byte-os startcímet kell kiválasztani, amely egyik területet sem zavarja, legcélszerűbb a BASIC-programterület elejét eltolni annyiszor 64 byte-tal, ahány sprite-nak kell terület.
2048 – eredetileg a BASIC-programterület eleje. (Ez pontosan 32*64). Eltoljuk ezt 64 byte-tal, így 2048+64=2112-t kapunk. Odatesszük a BASIC elejét:

```
POKE 43,(2112+1) AND 255
POKE 44,(2112+1)/256
POKE 2112,0
NEW
```

Ezután 2048-tól elhelyezzük a sprite adatait, végül POKE 2040,32-vel az 1. sprite címét beállítjuk a VIC-CHIP számára.

A többi sprite elhelyezésével és beállításával hasonlóképpen járjunk el, ide a második sprite rajzának a címét 2041-re, a harmadikat 2042-re tegyük és így tovább.

LEVÉLVÁLTÁS



Júniusi BIT-LET-ünkben közöltük Szolnoki Béla olvasónk hozzászólását a szoftverlopásokkal kapcsolatos korábban megjelent írásainkra. A témában egyelőre nem kaptunk újabb leveleket, hanem Szolnoki Béla egyik „megjegyzésére", levelet kapott az érintettől. A levél másolatát, majd Szolnoki Béla válaszát is megküldték az érintettek szerkesztőségünknek. Mi meg úgy gondoltuk, hogy érdemes a levélváltást napvilágra hozni. Íme:

Kedves Uram!
Az Ötlet 1986. június 26-i számának 29. oldalán lesújtó véleményt olvastam a DIGITEXT 64-ről az Ön hozzászólásában. Mint a program alkotója úgy éreztem, hogy erre reagálnom kell.
A hozzászólásban leírtakból sok mindennel egyetértek, én is furcsának tartom, például, hogy miért csak a külföldi programokból lett sláger és miért nem terjednek itthon (legalábbis hivatalos csatornákon) az eredeti magyar programok. Bár én nem panaszkodhatom, a DIGITEXT 64-ből a Novotrade elég jó forgalmat produkált annak ellenére, hogy az árát alacsonyabbra is megszabhatták volna.
A DIGITEXT 64-ről alkotott rossz véleményét nem egészen értem. Való igaz, hogy az első sorozat (10 példány) hibás volt, de aki a hibás lemezzel jelentkezett, azonnal megkapta a jó lemezt, sőt én magam vidékre is leutaztam kicserélni. A kezelési utasításhoz egy „Adatközlő lap"-ot mellékeltem, ezekből sajnos nagyon keveset kaptam vissza, így arra, hogy automatikusan cseréljem a még kintlévő esetleges rossz példányokat, adat hiányában nem volt módom.
Azt írja, hogy 5 percig használta a programot, a betöltési idő pedig háromszoros volt (azaz 15 perc?). Itt sajnos valami tévedés lehet, mert ennél lényegesen gyorsabb a betöltés. De azoktól, akikkel beszéltem (mint vásárlókkal) eddig még csak jó véleményt hallottam a programról, inkább azt, hogy kedvvel használják, mint azt, hogy elment tőle a kedvük.
Felmerült bennem az a kérdés, hogy vajon Ön eredeti DIGITEXT 64-et próbált ki? Ha igen, akkor miért nem kérdezte meg legalább telefonon (a kezelési utasításban a telefonszám benne van), hogy mit lehet tenni?
Az Ötletben leírt véleményét kicsit elhamarkodottnak tartom, de legfőképpen súlyos hitelrontásnak, mert 5 perces próba után gyakorlatilag használhatatlannak minősítette. Feltételezem, hogy nem hitelrontás és ellenpropaganda volt a célja, a hozzászólása nem ilyen alaphangulatú, azonban a leírt szónak a súlyát nem szabad figyelmen kívül hagyni.
Kérem ezért, hogy egyrészt legyen szíves belátása szerinti módon a program tényleges értékéről meggyőződni és ennek megfelelően korrigálni az elhamarkodott vélemény következményeit (pl. egy helyreigazítás formájában). Másrészt kérem, hogy az eredeti lemezt és az adatközlő lapot szíveskedjék elküldeni, hogy postafordultával cserélhessem, vagy a címét megadni, hogy személyesen elvégezhessem a cserét.
A levelem másolatát az Ötlet szerkesztőségének is elküldöm.
Budapest, 1986. június 27.
***dr. Bikfalvi István** 1124 Bp. XII., Pagony u. 8.*

Tisztelt Bikfalvi István!
Köszönettel vettem levelét, és igyekszem arra jól válaszolni, hogy Ön is helyre tudja tenni az Ötletben az Ön programjával kapcsolatos véleményemet. Talán annyit még, hogy vártam a reagálásokat és örülök, hogy a program szerzője megkeresett, talán néhány felhasználó is tollat ragad – bár eddig ez nem történt meg –, amely minden kétséget kizáróan bizonyítaná mások jó véleményét az Ön programjáról.
Mielőtt belebonyolódnék az Ön által föltett kérdések részletes megválaszolásába, jobbnak tartom, hogy tisztázzak néhány félreértést és leírjam azt a helyzetet, ahogy én a DIGITEXT 64-gyel megismerkedtem.
Bizonyára ismeri a BIT-LET Vallató rovatát. Nos, ismerőseim és barátaim számára én is szoktam néha „vallatni". Erre a Németországban szerzett diplomám, a nyelvtudásom és bizonyos számítástechnikai jártasságom ad alapot, tudniillik, hogy esetleges német vagy egyéb forrásból származó programokról véleményt alkossak. Nos, így kaptam egyik ismerősömtől is kipróbálásra egy lemezt, amelyen több szövegszerkesztő is volt, köztük a DIGITEXT 64 is.
A programot be is töltöttem a gépbe, majd rövid menüváltások után a karaktertervező részhez léptem, ahol a felszólításra, hogy nyomjam meg a kívánt billentyűt, lenyomtam a CTRL–E-t, mire a program lemerevedett. Amíg idáig jutottam, telt el a leírt kb. 5 perc. Ezután kikapcsoltam a gépet majd újra behívtam a programot, ezúttal egyenesen az előbb említett részhez ugrottam. Megismételve a műveletet ismét lemerevedett a program. Ekkor a folyamatot újra végigcsináltam, hasonló eredménnyel.
Ebből következik, hogy háromszor töltöttem be a programot, és hogy a betöltési időt nem számítva kb. 5 percet töltöttem el a program tesztelésével. De nem is dolgozhattam vele többet, hiszen – igaz véletlenül – rögtön a nekem odaadott program gyenge pontjára tapintottam. Ha erre csak egy óra múlva került volna sor, akkor annyi idő alatt ment volna el a kedvem. El kell ismernie, hogyha egy program ilyen tulajdonságokat mutat, elég az 5 perc is a véleménynyilvánításhoz.
Nem akarok ebbe most részletesebben belemenni, mivel jómagam is meggyőződtem arról, hogy a nekem véleményezésre adott program hibás volt. Én azonban ez alapján fogalmaztam meg a saját véleményemet. Túlzónak tartom ezért levelének „súlyos hitelrontás"-t emlegető sorait! Csakis az én személyes véleményemet írtam le, az pedig mindenki sajátja. Mivel azért gondoltam arra, hogy egy hibás program nem feltétlenül marad az, a leközölt levelemben oda is írtam, hogy („Persze lehet, hogy ez azóta már jobb lett."). Egyébként én a programról úgy tudtam, hogy az egy a Műszaki Egyetemen fejlesztés alatt álló szövegszerkesztő, nem pedig egy a kereskedelemben terjesztett program.
Mindezekből következik, hogy nem vagyok a program felhasználója, a program nincs és nem is volt a birtokomban, csakis egy véleményezés erejéig. Mindamellett azóta – még az Ön levele érkezése előtt! – személyesen győződtem meg arról, hogy van aki a jó DIGITEXT 64-et szereti és használja is. Ez azonban azután történt, hogy a levelemet elküldtem a BIT-LET-hez.
Remélem, hogy a leírtak az Ön számára is rávilágítanak a helyzetre, arra is, hogy eszembe sem jutott az, hogy a program általam nem is ismert alkotóját bántsam vagy a program ellen szándékosan szítsam a kedélyeket. A program a levelem megírásakor csak azért jutott eszembe, mivel beleillett annak tartalmába, de csak egy rövid megjegyzés erejéig, hiszen az ott a többhasábos nyomtatásban csupán 4 hasábsort foglal el. Sőt én inkább arra gondolok, hogy sokan, most hallanak először arról, hogy mégiscsak van egy magyar szövegszerkesztő.
Budapest, 1986. július 9. ***Szolnoki Béla***

Ami szerkesztőségünknek a dologról alkotott véleményét illeti. Már Szolnoki Béla levelének olvastakor fölmerült bennünk a kétely, hogy vajon javasoljuk-e a levélírónak, hogy ezt a bizonyos részt törölje leveléből. Beszéltünk vele a dologról, s mert meggyőzött bennünket, hogy ha a levél egyéb kérdésekben is az ő véleményét tükrözi, miért ne maradhatna benne az is. Nos, törzsolvasóink jól tudják, hogy alapelvünk a vélemények nyilvánításának teljes szabadsága, ennek szellemében adunk helyt a legszélsőségesebb véleményeknek is. (Természetesen csak egy darabig, mert nem kívánjuk olvasóinkat ismétlődő, előbbre nem jutó vélemények újra és újra való közlésével untatni.)
Nos, a levelet tehát változtatás nélkül közöltük. Nem is bánjuk különösebben. De ha már megnyilatkozásra késztet bennünket a fenti levélváltás, hát meg kell mondanunk, hogy nem értünk egyet Szolnoki Bélával. Legalábbis ami az ötperces tesztet illeti. Nem tudjuk, milyen hibája volt még az általa próbált Digitext-nek, de a felemlített hiba ellenére egy szövegszerkesztő lehetne jó. (A minálunk működő példányon egyébként ez a hiba nem jelentkezik.) Ami egyébként a Digitext-et illeti – nem rossz. Lehet vele dolgozni. (E sorok írója is dolgozott vele.) Van néhány kifejezett jó tulajdonsága, s vannak nagyon súlyos hiányosságai. Sajnos a leírás és a képernyőfeliratozás együttesen sem árul el egy csomó dolgot a felhasználónak. Ezekre a dolgokra (például printelési formátumok beállítása) vagy rájön az idők során a felhasználó, vagy sem. Sajnos azt kell mondanunk, hogy a Digitext az egyik legjobb hazai készítésű home kategóriájú gépre írott szövegszerkesztő. S ez a mondat már Szolnoki Béla általános mondandójának igazságát bizonyítja. S ráadásul azt is hozzá kell tennünk, hogy ez a minősítés nem annyira a Digitext nagyszerűségét, mint inkább versenytársai amatőrségét és hiányát bizonyítja. S ha már elárulta e sorok írója, hogy elég jól ismeri a Digitext-et, s hogy sokat dolgozott is vele, engedtessék meg, hogy az igazság kedvéért azt is hozzátegye, hogy mindez csak addig tartott, amíg rászánta magát, hogy megismerkedjen az Easyscripttel. Nos azóta a Digitext a „Kispadra" került.
A szerkesztő

TOKEN G2

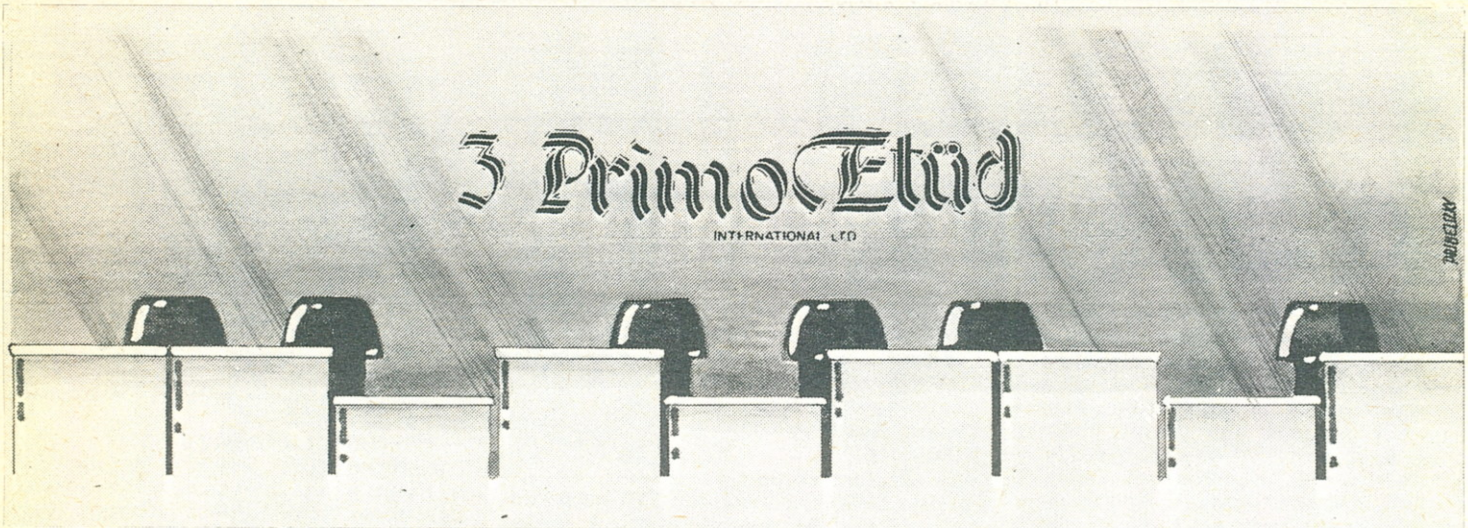
```
5 REM00000000000000000000000000000000000000000000000000000
10 L=PEEK(16548):H=PEEK(16549):A=L+256*H
20 POKE 16548,PEEK(A),PEEK(A+1):POKE 16424,L,H
50 POKE A,213,197,214,127,79,235,33,79,22,35,203,126,40,251,13,32,248,126,203,19
1,35,203,126,32,11,5,40,7,18,205,21,0,19,24,238,4,235,209,75,209,201
60 A=A+41
70 POKE A,33,40,64,54,L,35,54,H,195,0,1
80 H=INT(A/256):L=A-256*H
90 POKE 16415,L,H
95 CLS:PRINT"A program torlodik ,  de a kulcsszokiiras kikapcsolasig el !!":FOR
I=1 TO 2000:NEXT
1000 NEW
```

ROLL A és ROLL B

```
5 REM000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
10 L=PEEK(16548):H=PEEK(16549):A=L+256*H
20 POKE 16548,PEEK(A),PEEK(A+1)
30 Q=192:W=26
40 REM      Q:  a mozgatando sorok szama                    W: '26 -   LD A,(DE)
                    175 -    XOR A
45 REM               B a l r a
50 POKE 17390,42,73,64,46,0,229,209,35,6,Q,197,1,31,0,W,237,176,18,35,19,193,16,
243,201
55 REM               J o b b r a
60 POKE 17414,42,73,64,46,0,17,255,23,25,229,209,43,6,Q,197,1,31,0,W,237,184,18,
43,27,193,16,243,201
70 FOR I=1 TO 32:A=CALL(17390):NEXT
80 FOR I=1 TO 32:A=CALL(17414):NEXT
90 GOTO 70
```

ROLL G

```
5 REM0000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
10 REM000000000000000000000000000000000000000000000000000000000000000000000000000
0000
20 POKE 16548,16,69
90 REM....   Az  ablak  kijelolese   .....
100 DATA213,253,225,42,73,64,46,0,253,78,0,253,126,1,17,32,0,167,40,4,25,61,24,2
49,6,0,9,201
110 DATA62,32,253,150,2,95,22,0,253,70,3,201
120 DATA253,78,2,13,6,0,175,201
125 REM....   B a l r a   ...............
130 DATA205,237,67,205,9,68,213,229,209,35,197,205,21,68,237,176,18,193,209,25,1
6,240,201
135 REM....   J o b b r a  ...............
140 DATA205,237,67,253,126,3,253,78,2,205,251,67,175,237,82,205,9,68,43,213,229,
209,43,197,205,21,68,237,184,18,193,209,237,82,16,239,201
145 REM....   F e l        ...............
150 DATA205,237,67,205,9,68,5,197,213,229,209,1,32,0,9,205,21,68,12,237,176,209,
25, 1,32,0,237,66,193,16,232,0,229,209,19,205,21,68,119,237,176,201
155 REM....   L e          ...............
160 DATA205,237,67,253,126,3,253,78,2,205,251,67,175,237,82,205,9,68,5,43,197,21
3,229,209,175,1,32,0,237,66,205,21,68,12,237,184,209,175,237,82,1,32,0,9,193,16,
229,175,229,209,27,205,21,68,119,237,184,201
200 FOR I=17389 TO 17596
210 READ A:POKE I,A
220 NEXT I
225 X=5:Y=21:DX=20:DY=99
230 POKE 17600,X,Y,DX,DY
235 POKE 17605,5,21,20,50:POKE 17610,5,71,20,49
237 CLS:GOSUB 4000:PRINT#10,7,"G";CHR$(148);"      1985.dec.";
240 GOSUB 3000
250 FOR I=1 TO 20:A=CALL(17437,17600):FOR J=1 TO 20:NEXT J,I
260 GOSUB 3000
270 FOR I=1 TO 20:A=CALL(17460,17600):FOR J=1 TO 20:NEXT J,I
276 GOSUB 3000
278 FOR I=1 TO 100:A=CALL(17539,17600):NEXT
280 GOSUB 3000
290 FOR I=1 TO 100:A=CALL(17497,17600):NEXT
300 GOSUB 3000
310 FOR I=1 TO 50:A=CALL(17539,17605):A=CALL(17497,17610):NEXT
320 GOSUB 3000
330 FOR I=1 TO 50:A=CALL(17497,17605):A=CALL(17539,17610):NEXT
500 GOTO 240
3000 FOR I=0 TO 3:FOR J=0 TO49-14*I
3010 SET(70+14*I+J,121+J):SET(70+14*I+J,121-J):SET(168-J-14*I,121+J):SET(168-J-1
4*I,121-J):NEXT J
3020 NEXT I:RETURN
4000 FOR J=71 TO 171:SET(39,J):SET(200,J):NEXT
4010 FOR J=39 TO 199:SET(J,71):SET(239-J,171):NEXT:RETURN
5000 A$=CHR$(12)+CHR$(7)+CHR$(6)+CHR$(9)+CHR$(9)+CHR$(4)+" ROLL "+CHR$(20)+" G"+
CHR$(148)
5010 SAVE A$
```



Egy sokak által mellőzött számítógép, a PRIMO mellett szeretnék kiállni. Népszerűségének növekedéséhez talán sikerül hozzájárulnom a következő néhány programmal, ötlettel.

A Token G2 nevű program megírásához Gál Tamásnak a tavaly decemberi BIT-LET-ben megjelent programja adta az indíttatást. Lényegében ezt a programot egészítettem ki egy kis ötlettel. (A program tehát a kulcsszavak közvetlen képernyőre vitelét teszi lehetővé.) (Használatához táblázatot és ábrát már közöltünk lapunkban. – A szerk.)

Az eredetitől ez talán egy kicsit könnyebben kezelhető, mivel RESET-elés után nem kell újra és újra beírni azt az eredetileg megadott néhány sort. Itt a RESET miután használhatatlanná tette a programot, újra is éleszti.

A RESET-gomb benyomásakor ugyanis törlődik a 4027_H -címen található JP utasítás után általunk beírt ugrási cím. Ezt követően ugrás történik a RAM $401E_H$ címen található JP 100_H utasításra. Ez a cím tetszőlegesen átírható. Az itt megadott címen pedig elhelyezhető egy olyan szubrutin, ami a 4027_H címen található JP után beírja a nekünk megfelelő címet, és JP 100_H-zal záródik. Természetesen így a BASIC-program egyszeri lefutása után nincs többé szükség rá, így azt törli is az 1000 sorban található NEW utasítás. (Készséggel vállalom, hogy minden érdeklődőnek – kazetta és postaköltség biztosítása esetén – postafordultával elküldöm a program gépi kódú autostartos változatát.)

A programból hiányzó magyarázatok:
5 helyfoglalás
10–20 BASIC program kezdetének átállítása
70 RESET-eléskor erre a programra kerül a vezérlés, és beírja a nekünk megfelelő címet a 4027_H-en található JP után.
80–90 a $401E_H$ címen található JP 100_H utasítást írja át.

A ROLL A és ROLL B nevű programok, a képernyő tartalmának elforgatását teszik lehetővé.
A: visszaírással, B: törléssel, jobbra és balra.
A gépi kódú program egyszeri meghívása a képernyőtartalom 8 grafikus ponttal történő eltolását eredményezi. Így a képernyő elsötétítéséhez, vagy az egyszeri körbeforgatáshoz 32-szer kell meghívni.
A változóban megadhatjuk, hogy a képernyőtartalom hány grafikus pont szélességű sávját forgassa a program balra (felülről számítva), vagy jobbra (alulról számítva). Így pl. elérhető, hogy a képernyő felső 1/3-a balra, az alsó 1/3-a pedig jobbra mozdul el, míg a középső 1/3 helyben marad.
Q változóban megadhatjuk, hogy a Program A, vagy B típusú legyen. A közölt változat A típusú, hogy B legyen ahhoz a Q értékét kell megváltoztatni – láthatólag – 175-re. Nyilván ebben az esetben a 90-es sor is fölösleges – törlendő. A programok felépítéséről:
5 helyfoglalás
10–20 BASIC program kezdetének átállítása
50–60 a gépi kódú programot tölti be
Q: mozgatandó sorok száma
W: 26 – LD A, (DE) (a leeső byte-ot az A regiszterbe menti)
175 – XOR A (a regisztert nullázza)
Megjegyzés: mivel a két szubrutin (jobbra-balra: 17414, 17390) egymástól független, így Q és W értéke külön is beállítható. A programot begépelés után érdemes kazettán rögzíteni, mert egyszeri futtatás után a helyfoglaló sor törlődik. Futtatás után RUN 30-cal indítható újra.

A ROLL G fantázianevű program az előző program továbbfejlesztett változata. A program segítségével több (maximum a képernyő méretével megegyező) "ablakot" definiálhatunk és azok tartalmát a főirányokba eltolva (föl, le, balra, jobbra) sötétre változtathatjuk. Az ablak méret és helye 4 számmal egyértelműen leírható.
X: az ablak bal felső sarkának vízszintes koordinátája. (Értéke max. 29, min. 0 lehet)
Y: az ablak bal felső sarkának függőleges koordinátája. (Értéke max. 189, min. 0 lehet)
(A 0;0 pozíció a képernyő bal felső sarkában van.)
DX: az ablak szélessége.
(Értéke min. 2, max. értéke X-től függ.)
DY: az ablak magassága.
(Értéke min. 2, max. értéke Y-tól függ.)
(Y és DY mérőszámát a grafikus pontok száma adja.
DX-nek és X-nek megfelelő grafikus pontok száma pedig egyenlő 8∗DX, ill. 8∗X-el.)
A program betöltésekor a képernyő törlődik és a program nevének egy része inverzen jelenik meg. Ez annak köszönhető, hogy a PRIMO file névként olyan stringeket is elfogad, amik "speciális" karaktereket is tartalmaznak, így pl. elérhető, hogy a program betöltésekor az ernyő törlődjön, a programnév tetszőleges formátumban jelenlen meg, sőt a gép hangjelzést is adhat a betöltés megkezdésekor. Ennek a lehetőségnek csak az szab határt, hogy a név hossza max. 16 karakter lehet, a vezérlő karakterekkel együtt.
(Pl. 5000 A$=CHR$(12)+CHR$(6)+CHR$(7)+"ROLL":SAVE A$)
A program felépítéséről:
5– 10 helyfoglalás a gépi kódú rutinok számára
20 BASIC program kezdetének átállítása
90–160 DATA-sorok tartalmazzák a gépi kódokat
200–220 betöltés
225– X=5:Y=21:DX=20:DY=99.
230 POKE 17600, X, Y, DX, DY (az ablak definiálása)
.
.
.
250 balra 17437 (pl. A=CALL(17437, 17600))
270 jobbra 17460
278 le 17539
290 fel 17497
Az eltolás sebessége várakozó ciklusokkal lassítható. Erre akkor lehet szükség, ha különböző méretű ablakokat szeretnénk azonos sebességgel mozgatni. A vízszintes és függőleges mozgatás sebessége azonos méretű ablakok esetén is különböző. Figyelem! A ROLL A és B, valamint a ROLL G paraméterezési hibákat nem képes lekezelni!
A ROLL G esetén a paramétereket úgy kell megválasztani, hogy X+DX<=31, Y+DY>=191 legyen!

Geda Gábor, 3390 Füzesabony, Felszabadulás u. 16.

ötlet

PROGRAM AJÁNLAT

VC 20
F billentyűk programozása

Cikkeinkben gyakran hivatkozunk lapunk alapelvére, miszerint a szerkesztő azért van ... stb. Nos, ez az alapelv sokszor úgy érvényesül, hogy az olvasók által beküldött anyagok adják lapszámunk oldalainak egy részét, másszor meg úgy, hogy az olvasók kérdéseire adott válaszok, a kívánságok kielégítése tölt ki lapot, lapokat. Annak idején elhatároztuk, hogy a VC 20-as géppel nem nagyon foglalkozunk mert már „kiment a divatból", azután a hozzánk érkezett levelek, telefonhívások hatására megváltoztattuk véleményünket, s előbb vallatót, majd VC 20 prolongálva címmel cikkeket közöltünk a Commodore-ok doyenjével kapcsolatban. Elég régóta hallgatunk e gépről, s most egy ifjú olvasó kívánságát kielégítve ismét a VC 20-asok táborának kedvezünk. Az olvasó az alig 13 éves Melich Krisztián azt kérdezte tőlünk levelében, hogy hogyan lehetne programozni a gép F1–F8 billentyűit. Egy másik törzsolvasónk **Tóth Géza**, aki alig idősebb Krisztiánnál most rövid gépi kódú programmal válaszol a kérdésre. Ez a probléma ugyanis kicsit bonyolultabb, mint azt Krisztián hitte, efféle gépi kódú program nélkül ugyanis a dolog megoldhatatlan. Reméljük, a program más VC 20 tulajdonosoknak is hasznos, a gép rejtelmeibe beavatottaknak pedig tartalmaz némi programozói tanulságot.

A program bővített vagy bővítetlen VC 20-szal egyaránt használható. Betöltés után használata a következő: RUN-nal indítjuk, mire megjelenik a képernyőn egy kérdés:

Hány betűt tároljanak a billentyűk? (30–110)

Ez értelemszerűen azt jelenti, hogy a nyolc billentyű összesen 110 betűt képes tárolni és megjeleníteni. A válasz megadása után megjelenik a READY felirat és megkezdhetjük a billentyűk programozását. Ehhez az alábbi formátumot kell használnunk:

@ 1,"Tóth Géza programja"

A „kukac" mellett álló szám természetesen a billentyű száma. Ha netán a szöveg nem fér ki már (tehát túlléptük a megadott karakterszámot), akkor ILLEGAL QUANTITY hibajelzést kapunk, s próbálkozhatunk újra annak a billentyűnek a programozásával amelynél éppen tartottunk.

Ha elvégeztük a billentyűk programozását, akkor a következőképpen használhatjuk őket:

!ON – a programozott billentyűk bekapcsolása

!END – a programozott billentyűk kikapcsolása (de a RUN/STOP+RESTORE is ezt a hatást váltja ki)

Egy konkrét példa, amelyből egyéb érdekes lehetőségek is kiderülnek. Beírhatjuk például egy billentyűre az alábbit is:

@1,"LOAD"+CHR$(13)

Ha ezután !ON-t írunk, s megnyomjuk az F1-et, a gép úgy veszi mintha a LOAD parancsot írtuk volna a gépbe, s megnyomtuk volna a RETURN-t is, azaz elindul a töltés. (Hiszen a CHR$(13)= RETURN)

Ha a program beírásakor hibázunk, akkor futtatásnál „HIBA A PROGRAMBAN" hibajelzést kapunk. Ha az adatokban (DATA-kban) hibáztunk, akkor pedig ennek megfelelő hibajelzést.

A program működéséről:

A program miután megkérdezi a puffer méretét, kiszámítja mennyivel kell a BASIC felső határát lejjebb tolni, hogy a program és a puffer elférjen.

Az első FOR ciklus a gépi kódú programot tölti be (a "REM PROGRAM"-tól a "REM-ADATOK"-ig terjedő rész).

A "REM ADATOK" után következő DATA sorokban tároljuk a kezdőcímtől függő byte-ok relatív helyét a programban és a kezdőcímhez mért relatív tartalmukat. (Például JMP kezdőcím+580, a JMP után következő byte-okat úgy állítja be, hogy azok értéke K+580 legyen.)

```
0 PRINT"❏  ▓FUNKCIOBILLENTYUK"
1 PRINT
2 PRINT"HANY BETUT TAROLJANAK A BILLENTYUK?(30-110)
3 INPUTP
4 IFP<30ORP>110ORP<>INT(P)GOTO3
10 P=P+8:VR=PEEK(56)*256+PEEK(55)
20 VU=VR-P-385
30 VH=INT(VU/256):VL=VU-VH*256
40 K=VU+1:W=0
50 FORN=KTOK+395:READA:POKEN,A:W=W+A:NEXT:IFW<>49065THENPRINT"HIBA A PROGRAMBAN!
":END
55 W=0
60 READN:IFN=-1GOTO100
65 READA:C=K+A:CH=INT(C/256):CL=C-CH*256
70 POKEN+K,CL:POKEN+K+1,CH:W=W+A+N
80 GOTO60
100 IFW<>8886THENPRINT"HIBA AZ ADATOKBAN!":END
105 KH=INT(K/256):KL=K-KH*256:POKEK+350,KL:POKEK+352,KH
110 S=K+305:SH=INT(S/256):SL=S-SH*256:POKEK+370,SL:POKEK+372,SH
115 POKEK+282,P
116 POKE56,VH:POKE55,VL
120 POKE251,0:SYSK+369:NEW
130 REM---PROGRAM----
510 DATA165,153,240,3,76,29,242,165,211,133,202,165,214,133
520 DATA201,152,72,138,72,165,208,240,6,76,87,230,32,66,231
530 DATA165,198,133,204,141,146,2,208,13,164,251,240,243,185
540 DATA0,82,240,238,169,1,208,236,120,165,207,240,12,165
550 DATA206,174,135,2,160,0,132,207,32,161,234,32,99,80,201
560 DATA131,208,16,162,9,120,134,198,189,243,237,157,118
570 DATA2,202,208,247,240,194,201,13,208,187,76,29,230,164
580 DATA251,240,7,185,0,82,240,2,208,28,32,207,229,201,133
590 DATA144,24,201,141,176,20,233,132,201,4,176,1,24,42,41
600 DATA7,170,32,144,80,185,0,82,200,132,251,170,24,96,160
610 DATA255,138,240,9,200,185,0,82,208,250,202,208,247,200
620 DATA96,32,158,215,202,224,8,176,118,134,10,32,253,206
630 DATA32,158,205,32,163,214,133,147,170,240,102,48,100
640 DATA166,10,32,144,80,132,13,166,10,232,32,144,80,165
650 DATA13,56,101,147,133,183,32,236,80,169,0,133,251,166,183
660 DATA202,157,0,82,164,147,240,12,136,202,177,34,240,57
670 DATA157,0,82,136,16,245,96,132,97,133,98,162,8,32,144
680 DATA80,132,99,165,97,56,166,97,164,98,229,98,48,17,240
690 DATA14,228,99,176,10,189,0,82,153,0,82,232,200,208,242
700 DATA96,152,24,229,97,24,101,99,201,127,144,3,76,72,210
710 DATA168,166,99,202,228,97,144,233,189,0,82,153,0,82,136
720 DATA208,242,32,115,0,201,64,208,9,32,115,0,32,160,80
730 DATA76,174,199,201,33,208,9,32,115,0,32,85,81,76,174
740 DATA199,32,121,0,76,231,199,201,128,240,16,169,145,32
750 DATA255,206,162,1,160,80,142,36,3,140,37,3,96,162,14
760 DATA32,115,0,160,242,208,240,162,50,160,81,142,8,3,140
770 DATA9,3,96
781 DATA133,0,134,0,135,0,136,0,137,0,138,0,139,0,140,0
800 REM----ADATOK----
810 DATA69,98,132,143,188,143,196,143,206,235,242,143,316,159,329,340
820 DATA43,380,103,380,135,380,150,380,216,380,229,380,264,380,297,380,300,380,2
67,380
1000 DATA-1
```



NEM ELŐSZÖR, DE NEM IS UTOLJÁRA!
ÚJ HELYSZÍNEN! 1986. DECEMBER 13–14-ÉN
A MŰSZAKI EGYETEM KÖZPONTI ÉPÜLETÉNEK AULÁJÁBAN
BUDAPEST XI., MŰEGYETEM RAKPART 3–9.

Megközelítése: a tavalyi járművekkel (úgymint: repülőgép, bicikli, tengerjáró hajó), valamint tekintettel a Duna közelségére úszva, kajakozva.

Mindezeken kívül odavisz még a Keletitől a **7-es busz**, a Nyugatitól a **12-es busz**, a Déliből a **18-as villamos**. Autóparkolás az épület előtt minden mennyiségben.

Ajtónyitogatás: mindkét nap reggel 9-kor. Zárás: este 7-kor.

Belépő: DIÁKOKNAK, KATONÁKNAK ÉS GYEREKEKNEK 10 FORINT, MÁSOKNAK 20 FORINT.

PROGRAMCSEREBERE

Idén 50 géphelyet kínálunk a csereberélőknek. Mindegyikhez adunk tévét és csatlakozási lehetőséget. **A gépet, tárolót Önnek kell hoznia!** Egy asztal **egy órára 30 Ft-ért** bérelhető. A bérletek előjegyezhetők. Aki december 5-ig lefoglalja a helyet magának, az **20% árkedvezményt** kap. A helyfoglalást telefonon is, személyesen is intézhetik minden nap 9-től este 9-ig a Csokonai Művelődési Házban. **Telefon: 690-495** és **892-240.** A telefonos helyfoglalások alapján három napig tartjuk a megbeszélt géphelyeket. Ez idő alatt be kell fizetni a bérleti díjat személyesen vagy postán. **Postacím:** Czerny Zsuzsa – Csokonai Művelődési Ház 1153 Budapest XV., Eötvös u. 64–66.

PROGRAMBÖRZE

Egy másik helyiségben azok bérelhetnek asztalt, akik nem csereberélni akarnak, hanem saját készítésű programjaikat kívánják árusítani. Számukra 60 Ft/óra a bérleti díj. A programbörze teremben lévő asztalokra ugyanazok a bérlési módszerek érvényesek, mint a cserebererére.

JÓ BORNAK IS KELL A CÉGÉR!

A legjobb csereajánlat sem ér semmit, ha nem jut el az érintettekhez.

- Ezért hozta létre lapunk a programcserebere rovatot. E rovathoz ingyenes hirdetéseket veszünk föl a helyszínen, s ezeket később megjelentetjük a BIT-LET-ben!
- Ezért ajánljuk minimális térítésért az alábbi reklámeszközöket, amelyekkel bárki közzé teheti a rendezvény jellegéhez illő témájú hirdetéseit!

FÉNYÚJSÁG: 10 forintért vállaljuk, hogy az ön által megadott szöveget 5–10 alkalommal sugározzuk!

RÖPCÉDULA: ön megadja a szöveget, mi a kívánt példányszámban egy órán belül átadjuk önnek! A4-es laponként 1 forintért. A terjesztésben is szívesen segítünk!

HANGOS REKLÁM: ön kitalálja, mi bemondjuk egy tízesért!

SZENDVICS: azazhogy szendvicsembert is adunk, ha kell, vagy bármi más extra ötletének megvalósításában segítünk, az árban pedig megegyezünk.

BEMUTATÓK

Titokzatos terveink egyelőre még csak tervek, ezért a bemutatók végleges programját majd a december 11-i Ötletben olvashatják. **Terveink:** a **Macintosh** még mindig sláger; Eszik, vagy isszák az **Amiga-t** (bemutató és beszélgetés); Ablaktechnika a C64-en (avagy ilyen a **GEOS**); Hívjuk föl a legközelebbi adatbankot **(modembemutató)**; IBM kompatiblis-e az **IBM** kompatibilis? További bemutatóinkra külön kis pályázatot írtunk ki.

BESZÉLGETÉSEK

Szakértőink tanácsokat osztogatnak. Találkozhatnak a BIT-LET, a Commodore Újság szerkesztőivel.

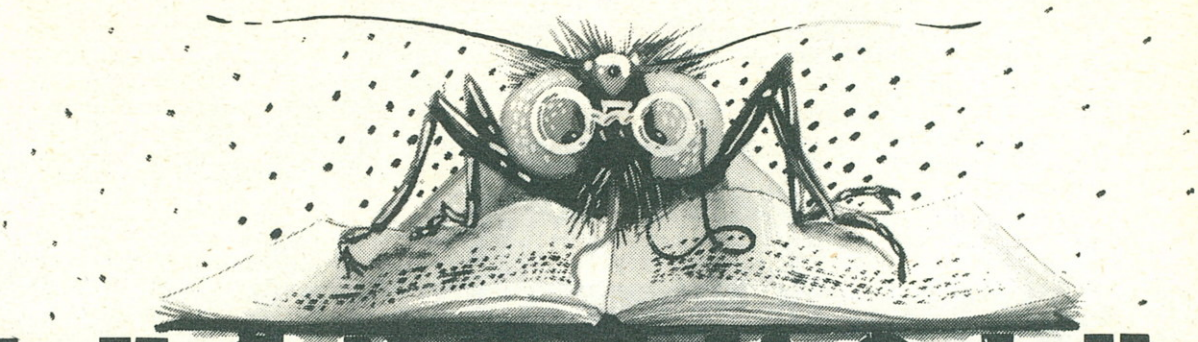
GARANCIÁK

Az idei BIT-LET Karácsony színvonalának garanciája, hogy védnökei a Novotrade és az ÁPISZ, rendezői a tavalyi sikeres rendezvény házigazdájának a a Csokonai Művelődési Háznak a munkatársai, a BIT-LET szerkesztősége, s mindehhez jön még idén a sok rendezvényen megedződött Műegyetemi Közművelődési Titkárság a maga stábjával.

SZENZÁCIÓ?

Tárgyalások folynak arról, hogy a BIT-LET Karácsony alkalmával is sugározna egész **Budapesten** fogható **rádióadó** számítógépes programokat!

öttlet



KÖNYVMOLY

Dr. Tokodi Jenő: **A LASER mikroszámítógépcsalád** – LSI ATSZ, 144 o., 140 Ft.
(A szerző – a LASER gépcsalád home-computer kategóriájú gépét véve alapul – a gépek BASIC nyelvének jellegzetességeit, gépi kódú programozásukat és a perifériák használatát ismerteti. A bemutatást kapcsolási rajzok, valamint a tokenek, és a rendszerváltozók táblázatai egészítik ki.)

Szilassy Bertalan: **C programozási útmutató** – LSI ATSZ, 58 o., 64 Ft.
(A dinamikusan fejlődő C programnyelv egységesítésére, hordozhatóvá tételére szolgáló javaslatok összefoglalása szabványajánlat formájában.)

Tudomány és technika Commodore 64 – DATA BECKER – Novotrade, 263 o., 215 Ft.
(A könyv a C 64 iskolai, egyetemi felhasználásához nyújt segítséget. Matematikai, természettudományos és műszaki feladatok megoldására mutat példákat, jó áttekintést adva a géppel megoldható problémákról.)

Varga József: Személyi számítógépek kezelése, programozása és alkalmazása – Terra, 326 o., 137 Ft.
Úgy tűnik, hogy a Terra kiadót elkerülték a mikroszámítógépek tömeges elterjedéséről szóló hírek.

A kötet címlapja ugyan ennek ellenkezőjét látszik igazolni, hiszen az a legismertebb géptípusok bemutatását ígéri (C–64, ABC–80, HT 1080Z, –2080Z, TEXAS-99, Primo–32, 64, TV COMPUTER–32, –64, VT–16). Ám belelapozva a könyvbe kiderül, hogy annak hangvétele, stílusa olyan, mintha a számítógépkezelés, a programírás még mindig egy szűk szakmai körre tartozna, mintha az ő privilégiuma volna. Ha ez így lenne, akkor a könyvet jó szívvel ajánlhatnánk a csekély számú vájtfülűnek, ők haszonnal forgathatnák.

Ezt a benyomást erősítik a száraz hangvételű, tudományoskodó magyarázatok is. A sok gépet bemutatva a szerző teljességre – vagy ahogy fogalmaz: az olvasó horizontjának szélesítésére – törekszik. Feleslegesen. Túl sokat markol, hiszen ezekhez a géptípusokhoz a felhasználónak rendelkezésére áll a szükséges, jóval részletesebb magyarázatokkal ellátott BASIC kézikönyv. Ilyen sok gépről ilyen kevés információra valószínűleg senkinek sincs szüksége. A rengeteg előre-hátra lapozást igénylő, bonyolult jelölésrendszer sem segíti a könyv használatát.

A kötet formája: kinyitva, jobb oldalon olvashatóak a C–64-re készült programrészletek, tudnivalók magyarázatokkal ellátva. Bal oldalon pedig a többi géptípusra vonatkozó módosítások, a C–64-től való eltérések felsorolása. Nos, a legrészletesebben bemutatott C–64-ről már számtalan könyv jelent meg, melyek – tárgyuk egységessége miatt – alaposabban mutatták be a gép kezelését, programozását. Így a Commodore használói nem sok újat tudhatnak meg a kötetből. A többi gép kezelői pedig még kevesebbet, hiszen a nekik szóló néhány soros, túltömörített megjegyzések szinte áttekinthetetlenek.
Hol lehet mégis hasznos a kötet? Talán az oktatásban, hiszen a szerző maga is tankönyvnek nevezi munkáját.
Nézzük először az önálló tanulást! Mit sajátíthat el a könyvből az olvasó segítség nélkül? Az első fejezet negyedik oldaláról egy példaprogram:

```
10 READ A$
20 DATA KIS ILONA
30 PRINT A$
40 END
```

Ehhez a magyarázat: „Programunkban kulcsszavak: READ (olvas), DATA (adat), PRINT (nyomtat), END (vége), változó: A$ és adat: KIS ILONA szerepelnek. Az A változó valós számokat, a $ jellel kiegészítve szövegeket (string) azonosíthat. A program végrehajtását a RUN (fut) parancs indítja el. Az A$ változóhoz hozzárendeli a DATA készletét (A$= KIS ILONA), majd képernyőre írja A$ értékét. A programok END-del zárulnak." Hát, aki a géppel először találkozva megérti, hogy mi is történik – annak nincs is szüksége arra, hogy könyvből tanuljon programozni.
Hol lehet mégis hasznos a kötet? Esetleg, mint az iskolai számítástechnika-oktatás segédkönyve? Ismét egy rövid programrészlet:

```
10 PRINT "SZAMITHATO: FELKAMATOLT TOKE, DISZKONTALT TOKE, JARADEKOSSZEG, JARADEKTAG"
15 PRINT "KOLCSONOSSZEG, TORLESZTOTAG"
...
```

Ehhez hozzáfűzni sem kell semmit.
Nem tudni, hogy a szerzőtől vagy a kiadótól származik ez a szerkesztési koncepció – illetve ez a koncepciótlan szerkezet. Varga József a terjedelmi keretek között mindent megtett, hogy bemutassa a gépeket – de ez egyetlen köteten belül, az összes elterjedt géptípussal foglalkozva megoldhatatlan feladat. A bevezetőben felajánlja az olvasóknak, hogy felmerülő problémáikra levélben válaszol. Félő, hogy a közeljövőben jelentősen meg fog nőni a levelezése.

Dachsel: Zenekönyv a Commodore 64-eshez – Data Becker-Novotrade, 151 o., 323 Ft.
A kötet leendő olvasója azt hihetné, hogy ez a könyv egyaránt szól a zenéhez értőkhöz (zenészekhez) és a számítástechnikával foglalkozókhoz. Vagy, ha mindkét csoporthoz nem is, legalább az egyikhez. Hamarosan kiderül azonban, hogy – sajnos – egyik olvasói tábor sem zárhatja szívébe a szerzőt.
A bevezető még sok szépet és jót ígér. Hosszú (és tartalmas!) eszmefuttatás olvasható itt arról, hogy a „zene" fogalma, illetve az, hogy mely hangok egymásutánját és egyidejű hangzását tekintjük zenének, menynyiben függ az adott kortól, kulturális környezettől, divattól. A leírtak mind helytállóak, az elméleti megalapozással nem lehet problémánk. Nyilvánvaló a végkövetkeztetés is: a hanggenerálásra alkalmas számítógép is szolgáltathat zenét. (Bár az a megjegyzés erősen megkérdőjelezhető, mely szerint a C–64-be épített hangchip felér egy szintetizátorral.)

A gondok akkor kezdődnek, amikor a szerző mindezt a gyakorlatba próbálja átültetni. Már a hatodik kis mintaprogram is tömve van a (nem profi) programozó számára értelmezhetetlen DATA-sorokkal. Indoklás: „Ez a rövid BASIC-program megmutatja minden zeneprogram alapvető szerkezetét, mely a következő: egy rövid programrészlet (0–6-os sor), és egy igen hosszú adatsor (DATA-sorok 7-től 16-ig)." Az olvasó kénytelen elfogadni ezt a szerkezetet, remélve, hogy később többet is megtud a DATA-sorok adatairól.

A szerző alapállása itt válik ellentmondásossá. Egyrészt olyan tudást tételez fel – mind a zene, mind a programozás terén –, ami korántsem nyilvánvaló. Másrészt szájbarágós – de átgondolatlan – példákkal igyekszik megvilágítani a korábban már ismertnek feltételezett fogalmakat.

Így például a hangchip működését egy meglehetősen ködös metaforával írja le, mely szerint számozott irodaszobákban ülő alkalmazottak engedelmeskednek a (szintén számokkal jelölt) utasításoknak. A leírás a POKE utasítás hatását volna hivatott bemutatni. Ennek lényegét a programozást egy kicsit is ismerő olvasó úgy is érti. Azok számára viszont, akik a zene miatt vették kezükbe a kötetet, és valóban szükségük lenne jó magyarázatokra, a szerző egy fél sort sem veszteget például a FOR – NEXT ciklus működésének leírására – pedig ebből már a legelső programokban is nem egy található.

Hasonló módszerrel folytatódik a zeneprogram írásának bemutatása: listázzuk ki a megadott sort, javítsuk ezt a számot arra, emezt meg amarra, és lám! a gép kétszer olyan gyorsan játsza le a dallamot. De hogy miért?...

A szerző az egyszólamú, BASIC nyelvű dallamok megszólaltatásától végül is eljut az assemblerben írt, többszólamú zeneprogramokig – ugyanilyen tárgyalásmóddal. Mindezt rengeteg táblázat, és igen tömény hangtechnikai leírás kíséri.

A zeneprogramozással most ismerkedő olvasó csak igen körülményesen, sok-sok önálló kísérletezéssel, és más szakkönyvek párhuzamos böngészésével használhatja csak a kötetet. Azok számára, akik már jól ismerik a hangchip programozását, hasznos lehet kézikönyvként. Részükre viszont elegendő – és célszerűbb is – lett volna egy összefoglaló táblázat- és információgyűjtemény kiadása a laikusokat megtévesztő körítés nélkül.

Tallér József

A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!

POSTA

Van „nekem" egy Commodore VC 20 típusú számítógépem. Egyszerűen nem tudok vele rajzolni. A grafikus karaktereket ismerem, de annál bonyolultabb ábrát nem tudok kirajzolni a képernyőre. Hogy ez lehetséges legyen, kérem, hogy vagy az ÖTLET-ben, vagy esetleg levélben VC–20-hoz mutassanak rajzoló programokat. Előre is köszönöm.

Tóth Péter, 4. oszt. tanuló, Debrecen, Jerikó u. 32. IV/32.

Javasoljuk, hogy tanulmányozd át a BIT-LET 1985. márciusi számában megjelent ilyen témájú cikkünket. (Bármelyik nagyobb könyvtárban megtalálod!)

Murányi Szabolcsnak hívnak, 13 éves vagyok, 2 éve foglalkozom számítástechnikával. Gondoltam, ennyi év után miért ne írhatnék én is egy játékprogramot. Bele is vágtam és már az utolsó igazításokat végzem rajta. A játék neve vagy Labirintus vagy Ments meg hős királyfi lesz. De a játékhoz szükségem lenne egy 25∗30-as háttérre a 25∗40 helyett. (A játék egyébként C 16-oson készül.) Azt szeretném megkérdezni, hogy hogyan lehet ezt elérni programban és a program végén ezt hogyan lehet megszüntetni és visszaállítani a 25∗40-es háttért?
Továbbá szeretném megtudni, hogyan lehet POKE-val elérni a magyar ékezetes betűket és van-e ASCII kódjuk, ha nem lehet PKE-val és ASCII kóddal elérni őket, kérem küldjenek nekem egy olyan programot, ami magyar ékezetes betűket állít elő.

Murányi Szabolcs, *2400 Dunaújváros, Váci M. u. 11. XI/4.*

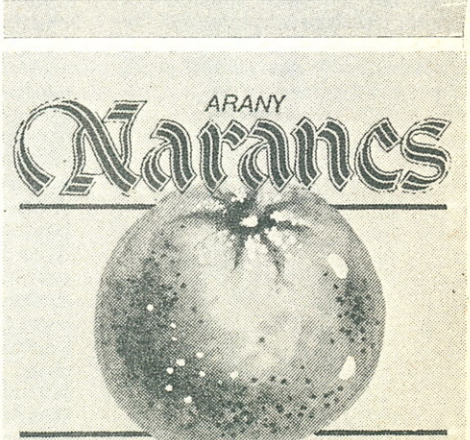
Az ESC-funkciók segítségével jelölhetünk ki ablakokat a képernyőn. Így csinálhatunk pl. 25∗30-as képernyőt a 25∗40-es helyett:

```
10 CHAR1,35,24,CHR$(27)+"B"   :REM ESC"B"
11 PRINT"▒▒▒▒▒▒"+CHR$(27)+"T":REM ESC"T"
20 DO:PRINT"A";:LOOP
25 :
30 REM NORMAL KEPERNYO:25X40-ES=ESC"N"
```

ESC"T" = Az ablak felső sarkának kijelölése.
ESC"B" = Az ablak alsó sarkának kijelölése.

Ami az ékezetes betűket illeti: Csak a speciálisan „előkészített" C 16-osokon vannak ékezetes betűk. Ezekre a gépekre rá is égették a megfelelő billentyűkre a megfelelő betűket. Nyilván, hogy az egyes ékezetes karaktereknek ezeken a gépeken ASCII kódjuk is van. Ékezetes betűket generáló szoftvert lehet kapni a C 16-oshoz a 2C üzletekben. Sajnos mi nem foglalkozunk programok küldésével.

A NARANCS DÍJ-at eddig még nem adtuk ki. Most azért döntöttünk úgy, hogy kiadjuk, mert az alábbi apró felfedezést megmutattuk szinte minden C16-tal foglalkozó ismerősünknek, barátunknak, s mindannyiuk számára meglepő, új és izgalmas volt az. A narancs díj az erkölcsi dicsőségen túl a közölt anyag terjedelmével arányban nem álló „extra" honoráriumot jelent a fiatal szerző számára.



RE-NEW Commodore 16-ra

A BIT-LET 1986. áprilisi számában a Hasznos apróságok című cikkben megjelent már egy eljárás a NEW paranccsal vagy a RESET-tel kitörölt programok visszahozataláról. A most ismertetésre kerülő eljárás azonban jóval egyszerűbb, és külön előnye, hogy a sorszámokat sem változtatja meg.

A program visszahozatala a
POKE 4097,1 : RENUMBER
paranccsal történhet. Az első rész az \$1001–\$1002 címeken található mutató értékét (mely egyébként a 2. BASIC sor címére mutat) állítja nem zérus értékűre. A RENUMBER parancs hatására pedig az interpreter a megfelelő értékre állítja be a BASIC rendszerváltozókat, de a program sorszámozását nem változtatja meg. Ez különösen akkor hasznos, ha véletlenül fejlesztés közben törlődött a program pl. egy hibásan működő gépi kódú szubrutin miatt.

Grósz Attila 5600 Békéscsaba, Lencsési u. 8.

HARMADGÉPNYERŐ, AZ 1. FELADAT MEGOLDÁSA:

A feladat négy részből áll. Íme ezek megoldásai sorrendben:

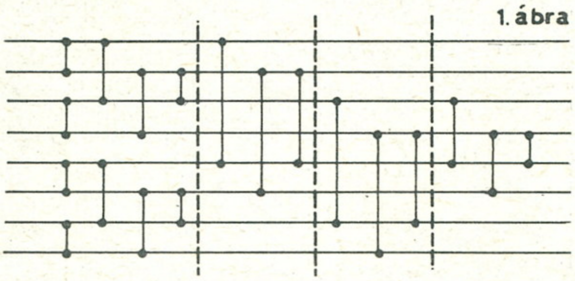
1. Tegyük fel az állítással ellentétben, hogy van egy olyan network, mely minden 0–1 sorozatot rendez, de az 1, 2, ..., n számoknak van egy olyan permutációja, melyet nem rendez, azaz egy nagyobb sorszámú dróton kisebb érték (jelöljük ezt az értéket m-mel) érkezik, mint egy kisebb sorszámún. A networkök működéséből nyilvánvaló, hogy ha az 1, 2, ..., m számok helyett 0-t, az m+1, m+2, ... n számok helyett 1-et engedünk rá a networkre, akkor az m db 0 pont oda fog érkezni, ahova eredetileg az 1,2, ..., m számok érkeztek, az egyesek pedig az m-nél nagyobb számok helyére fognak érkezni.

Az eddigieket egybevetve láthatjuk, hogy most egy kisebb sorszámú drótra 1-es fog érkezni, míg egy nagyobb sorszámúra (az m helyére) 0-s. Ez viszont ellentmond annak, hogy a network minden 0–1 sorozatot rendez. Feltételünkből ellentmondásra jutottunk, ezért az állítás igaz. (Megjegyzés: ezt a bizonyítási módszert, amikor az állítás tagadásának lehetetlenségét bizonyítjuk, indirekt bizonyítási módszernek nevezzük. A bebizonyított állítás neve pedig 0–1 törvény.)

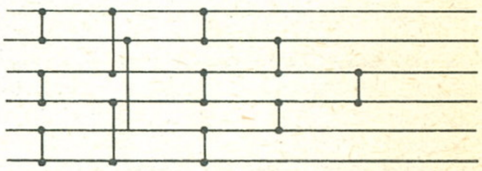
2. Azt, hogy a 2. ábrán látható kisebb network valóban mindig rendez 4 számot, a 0–1 törvény segítségével könnyen láthatjuk. A 3. ábrán szerepelt networköt 4 részre bonthatjuk a mostani 1. sz. ábra szerint. Az első rész rendezi külön az alsó 4 és a felső 4 dróton érkező számokat. A másik 3 rész mind szintén a 4-es rendező, de az első 2 kapu kivételével. Nézzük először a 2. részt! Mivel ekkor tudjuk, hogy ide a számok már úgy érkeznek, hogy a 2. dróton >=szám van, mint az 1-n, és a 6-on is >= van, mint az 5-en, így ezt a 4 drótot (1, 2, 5, 6) ez a rész már egymás közt rendezni fogja. Emiatt a két legkisebb szám a helyére kerül. Hasonlóan a 3. rész rendezi egymás közt a 3, 4, 7 és 8. drótokat, így a két legnagyobb is a helyére kerül. Ezenkívül a 2. rész gondoskodik arról, hogy az 5. dróton <=szám legyen, mint a 6-on; a 3. rész pedig arról, hogy a 3-on legyen <=, mint a 4-en, így a 4. részt rendezni fogja a 3, 4, 5, és 6. drótokat, s ezzel a rendezést teljessé teszi. (Megjegyzés: a 2, 3 és 4. rész működését leellenőrizhetjük a 0–1 törvény segítségével is, az 1. rész működésének ismeretében csak 25 esetet kell ellenőrizni.)

3. Az állítás nyilvánvaló, hiszen, ha az 1, 2...n számokat pont sorban adjuk be, akkor ugyanúgy is fognak megérkezni (minden kapuhoz úgy érkeznek a számok, hogy a kisebb sorszámú dróton kisebb szám érkezik, így egyik kapu se cserél!), tehát ekkor biztos, hogy a kisebb sorszámú drótra kisebb szám érkezik!

4. Lásd a 2. ábrát! A működés könnyen ellenőrizhető a 0–1 törvény segítségével, figyelembe véve a szimmetriát, elég azokat az eseteket kipróbálni, amikor maximum 3 db 0-s van. Házi feladat (nem beküldésre): írjunk olyan programot, mely szimulálni tudja egy tetszőleges network működését, s a 0–1 törvény segítségével leellenőrzi, hogy rendező-e!

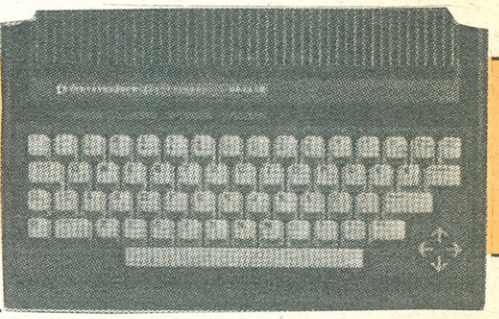


1. ábra



2. ábra

PLUS/4 NYERŐ



1. FELADAT

Új pályázatunk három hónapos, s díja egy COMMODORE PLUS 4-es gép. A következő három hónapban közölt három feladatban legtöbb pontot szerzők közt súlyozott módszerű sorsolást tartunk majd. Az első feladattal szerezhető maximális pontszám 70 pont.

Sok gyakorlati probléma megoldásánál merül fel, hogy egy úgynevezett gráfon 2 pont között legrövidebb utat kell keresni (pl. a C 16 nyerő 2. feladatában, értékelését l. ebben a számban). A gráfok olyan „dolgok", ahol van néhány pont, amiket csúcsnak nevezünk, s minden lehetséges pontpárra meg van adva, hogy össze vannak-e kötve, vagy nem. Az első esetben azt mondjuk, hogy a két csúcs között megy él, a második esetben azt, hogy nem megy. Ezenkívül megengedjük, hogy az élekre egy-egy szám legyen írva, ami pl. a két pont közti út hosszát, vagy valami hasonlót jelképez. A legrövidebb út keresése azt jelenti, hogy kijelölünk egy kezdő- és egy végpontot, s el akarunk jutni a kezdőpontból a végpontba végig élek mentén úgy, hogy az út által érintett élekre írt számok összege a lehető legkisebb legyen.

Legrövidebb út keresésére való a következő, Dijsktrától származó algoritmus:

Ez az algoritmus egy előre kijelölt kezdőpontból minden más pontba keresi a legrövidebb utat. A csúcsokat címkézni fogjuk, mégpedig kétféle címke lesz: végleges és kísérleti. A végleges címkére mindig az odavezető legrövidebb út hossza (az élekre írt számok összege) lesz írva, míg a kísérleti címkék csak felső korlátot adnak ugyanerre.

0. lépés: **a kezdőpont kapjon egy végleges címkét, melyre 0-t írunk. Az összes többi csúcs kapjon egy kísérleti címkét, melyre a kezdőpontból hozzávezető élre írt számot írjuk, ha van ilyen él, különben +∞-t. (Megjegyzés: számítógépes megvalósítás esetén a gép által ábrázolható számok közül a legnagyobbat szokás ide írni.)**

1. lépés: **Keressünk egy olyan kísérleti címkés csúcsot, melyre a lehető legkisebb szám van írva. (Ha több ilyen van, ezek közül válasszunk ki egyet!) Ennek a csúcsnak a címkéjét véglegesítjük úgy, hogy ugyanaz maradjon ráírva. Ha nincs több kísérleti címkés csúcs, álljunk meg.**

2. lépés: **Módosítsuk a kísérleti címkéket! Ha a most véglegesített csúcsból vezet él valamely kísérleti címkés csúcsba, s az erre az élre írt szám és a most véglegesített címkére írt szám összege kisebb a kísérleti címkére írt számnál, akkor ezt az összeget írjuk a kísérleti címkére. (Ezt minden kísérleti címkés csúcsnál meg kell vizsgálni!) Menjünk az 1. lépésre!**

A feladat az, hogy

1. **Bizonyítsuk be: ha minden élre pozitív szám van írva, akkor Dijkstra algoritmusa jól működik, tehát leálláskor minden csúcsnak a címkéjén valóban az odavezető legrövidebb út hossza lesz (ill. +∞, ha nem lehet a kezdőpontból az illető csúcsba eljutni).**

2. **Az algoritmus végrehajtása során hány összeadásra és hány összehasonlításra van szükség? (A gráfnak n csúcsa van. Elég körülbelül is!)**

3. **Mit kell tennünk, ha nemcsak a legrövidebb út hosszára, hanem magára az útra is kíváncsiak vagyunk? Próbáljuk úgy módosítani az algoritmust, hogy végrehajtása után ezt is le lehessen olvasni valahonnan. Mennyivel növeli ez a lépésszámot?**

Kérjük levágni és a levélre felragasztani! Beküldési határidő: 1987. január 10.

Ápisz

BIT-LET

Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál
Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Gépnyerő: Király Zoltán. Posta: Halász Péter